On the Adaptive and Learning Control Design for Systems with Repetitiveness

BY

DEQING HUANG

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2010

Acknowledgments

I would like to express my deepest appreciation to Prof. Xu Jian-Xin for his inspiration, excellent guidance, support and encouragement. His erudite knowledge, the deepest insights on the fields of control have been the most inspirations and made this research work a rewarding experience. I owe an immense debt of gratitude to him for having given me the curiosity about the learning and research in the domain of control. Also, his rigorous scientific approach and endless enthusiasm have influenced me greatly. Without his kindest help, this thesis and many others would have been impossible.

Thanks also go to Electrical & Computer Engineering Department in National University of Singapore, for the financial support during my pursuit of a PhD.

I would like to thank Dr. Lum Kai Yew at Temasek Laboratories, Prof. Zhang Weinian at Sichuan University, and Dr. Qin Kairong at National University of Singapore who provided me kind encouragement and constructive suggestions for my research. I am also grateful to all my friends in Control and Simulation Lab, the National University of Singapore. Their kind assistance and friendship have made my life in Singapore easy and colorful.

Last but not least, I would thank my family members for their support, understanding, patience and love during past several years. This thesis, thereupon, is dedicated to them for their infinite stability margin.

Contents

Summary

The control of dynamical systems in the presence of all kinds of repetitiveness is of great interest and challenge. Repetitiveness that is embeded in systems includes the repetitiveness of system uncertainties, the repetitiveness of control processes, and the repetitiveness of control objectives, etc, either in the time domain or in the spatial domain. Learning-type control mainly aims at improving the system performance via directly updating the control input, either repeatedly over a fixed finite time interval, or repetitively (cyclically) over an infinite time interval. In this thesis, the attention is concentrated on the analysis and design of two learning-type control strategies: adaptive control (AC) and iterative learning control (ILC), for dynamic systems with repetitiveness.

In the first part of the thesis, two different AC approaches are proposed to deal with nonlinear systems with periodic parametric repetitiveness in continuous-time domain and in discrete-time domain respectively, where the periodicity could be temporal or spatial.

Firstly, a new spatial periodic control approach is proposed to deal with nonlinear rotary machine systems with a class of state-varying parametric repetitiveness, which is in an unknown compact set, periodic, non-vanishing, and the only prior knowledge is the periodicity. Unlike most continuous time adaptation laws which are of differential types, in this work a spatially periodic type adaptation law is introduced for continuous time systems. The new adaptive controller updates the parameters and the control signal periodically in a pointwise manner over one entire period along the position axis, in the sequel achieves the asymptotic tracking convergence.

Consequently, we develop a concise discrete-time adaptive control approach suitable for

nonlinear systems with periodic parametric repetitiveness. The underlying idea of the new approach is to convert the periodic parameters into an augmented constant parametric vector by a lifting technique. As such, the well-established discrete-time adaptive control schemes can be easily applied to various control problems with periodic parameters, such as plants with unknown control directions, plants in parametric-strict-feedback form, plants that are nonlinear in parameters, etc. Another major advantage of the new adaptive control is the ability to adaptively update all parameters in parallel, hence expedite the adaption speed.

ILC, which also can be categorized as an intelligent control methodology, is an approach for improving the transient performance of systems that operate repetitively over a fixed time interval. In the second part of the thesis, the idea of ILC is applied in four different topics under the repetitiveness of control processes or control tasks.

As the first application, an initial state ILC approach is proposed for final state control of motion systems. ILC is applied to learn the desired initial states in the presence of system uncertainties. Four cases are considered where the initial position or speed are manipulated variables and final displacement or speed are controlled variables. Since the control task is specified spatially in states, a state transformation is introduced such that the final state control problems are formulated in the phase plane to facilitate spatial ILC design and analysis.

Then, a dual-loop ILC scheme is designed for a class of nonlinear systems with hysteresis input uncertainty. The two ILC loops are applied to the nominal part and the hysteresis part respectively, to learn their unknown dynamics. Based on the convergence analysis for each single loop, a composite energy function method is then adopted to prove the

learning convergence of the dual-loop system in iteration domain.

Subsequently, the ILC scheme is developed for a class of nonlinear partial differential equation processes with unknown parametric/non-parametric uncertainties. The control objective is to iteratively tune the velocity boundary condition on one side such that the boundary output on the other side can be regulated to a desired level. Under certain practical properties such as physical input-output monotonicity, process stability and repeatability, the control problem is first transformed to an output regulation problem in the spatial domain. The learning convergence condition of iterative boundary learning control, as well as the learning rate, are derived through rigorous analysis.

To the end, we propose an optimal tuning method for PID by means of iterative learning. PID parameters will be updated whenever the same control task is repeated. In the proposed tuning method, the time domain performance or requirements can be incorporated directly into the objective function to be minimized, the optimal tuning does not require as much the plant model knowledge as other PID tuning methods, any existing PID auto-tuning methods can be used to provide the initial setting of PID parameters, and the iterative learning process guarantees that a better PID controller can be achieved. Furthermore, the iterative learning of PID parameters can be applied straightforward to discrete-time or sampled-data systems, in contrast to existing PID auto-tuning methods which are dedicated to continuous-time plants. Thus, the new tuning method is essentially applicable to any processes that are stabilizable by PID control.

List of Figures

List of Tables

Nomenclature

Symbol	Meaning or Operation
$\forall$	for all
$\exists$	there exists
$\triangleq$	definition
$\in$	in the set
$\subset$	subset of
$\mathbf{N}$	set of integers
$\mathbf{R}$	set of real numbers
$\mathcal{S}(\star)$	signum function
$\mid \star \mid$	absolute value of a number
$\| \star \|$	Euclidean norm of vector or its induced matrix norm
$\mathcal{A}$	fixed initial speed in initial position learning
u_x	initial position input at $\mathcal{A}$
u_v	initial speed input at a fixed initial position
x_f	prespecified final position in speed control
v_f	prespecified final speed in position control
x_d	desired final position
v_d	desired final speed
$x_e(u)$	observed final position in position control
$v_e(u)$	observed final speed in speed control
$u_{x,i}, u_{v,i}$	inputs in the i-th iteration
$x_{i,e}, v_{i,e}$	observed outputs in the i-th iteration
$x(v,u), v(x,u)$	solutions trajectory with input u

Symbol	Meaning or Operation
f_k	a time function $f(k)$ or a function $f(x_k)$ w.r.t. its argument x_k
$\hat{\theta}$	the estimate of θ
$\tilde{\theta}$	$\tilde{\theta} = \theta - \hat{\theta}$
$\mathbf{x}$	vector with certain dimensions
$\mathcal{C}[0,T]$	continuous function spaces in $[0,T]$
$\mathcal{C}^1[0,T]$	continuously differentiable function spaces in $[0,T]$
AC	Adaptive Control
CM	Contraction Mapping
ES	Extremum Seeking
ZN	Ziegler-Nicholes
CEF	Composite Energy Function
IFT	Iterative Feedback Tuning
ILC	Iterative Learning Control
ILT	Iterative Learning Tuning
IMC	Internal Model Control
ISE	Integrated Square Error
ISL	Initial State Learning
LKF	Lyapunov-Krasovskii Functional
ODE	Ordinary Differential Equation
PAC	Periodic Adaptive Control
PDE	Partial Differential Equation
PPT	Pole-Placement
IBLC	Iterative Boundary Learning Control
ISIO	Incrementally Strictly Increasing Operator
SISO	Single Input Single Output
SPAC	Spatial Periodic Adaptive Control

Chapter 1

Introduction

1.1 Learning-type Control Strategies and System Repetitiveness

The control of dynamical systems in the presence of all kinds of repetitive uncertainties is of great interest and a challenge. Among existing control methods, learning-type control strategies play an important role in dealing with systems with repetitive characteristics. These methods include adaptive control, repetitive control, iterative learning control, neural networks, etc. In fact, learning can be regarded as a bridge between *knowledge* and *experience* [29]. In control engineering, *knowledge* represents the modelling, environment, and related uncertainties while *experience* can be obtained from the previous control efforts, and some resulting errors through system's repetitive operations.

Investigating the learning behavior of human beings, a person learns to know his/her living environment from the daily activities, and acquires knowledge through the past events for future actions. In the learning process, similar or same activities occur again and again, hence the inherent and relevant knowledge also repeats. Thus, repetitiveness is always a key point to any successful learning of human beings. Similarly, the systems considered with learning-type control strategies should at least take some repetitiveness,

including repetitiveness of system uncertainties, repetitiveness of control processes, and repetitiveness of control objectives, etc. In the next, let us address the three kinds of repetitiveness separately.

(1) Repetitiveness of system uncertainties. This class of repetitiveness refers to the periodic invariance of parametric components, non-parametric components, and external disturbances since periodic variations are invariant under a shift by one or more periods. They are often a consequence of some rotational motion at constant speed, and encountered in many real systems such as electrical motors, generators, vehicles, helicopter blades, and satellites, etc. These uncertainties may be periodic in the time domain or the spatial domain, and the period is usually assumed to be known and stationary. Obviously, constant unknowns in system should also belong to this category.

(2) Repetitiveness of control processes. Here, we usually consider the processes that repetitively perform a given task over a finite period of time. Thus, every trial (cycle, batch, iteration, repetition, pass) will end in a fixed time of duration. In a strict point of view, invariance of the system dynamics, repetition of outer disturbances, and repetition of the initial setting must be ensured throughout these repeated iterations. It is worth noticing that different from the repetitiveness in scenario (1), repetitiveness of control processes is often demonstrated in the iteration domain, instead of the time or state domain.

(3) Repetitiveness of control objectives. In many learning-type control objectives, the desired output/input trajectory periodically varies in an infinite time horizon. Thus, the control objective shows the repetitiveness with a periodicity in the time domain. Notice that the control process for this scenario may not show any repeatability.

In practice, system repetitiveness could be a combination of the above three types

of repetitiveness, or more other repetitiveness that is not mentioned here. For instance, a robotic manipulator consecutively draws a circle in Cartesian space with the same radius but different periods, or on the contrary, draws the circle with the same period but different radii. Although non-repetitiveness is contained in control objectives and control processes, repetitiveness still exists as a main characteristic in them.

Corresponding to different repetitive environment, learning control methods exhibit different learning procedures. For instance, AC [60, 67, 101] is a technique of applying some system identification techniques to obtain a model of the process and its environment from input-output experiments and using this model to design a controller. The parameters of the controller are adjusted during the operation of the plant as the amount of data available for plant identification increases. AC is good at the control of systems with parametric repetitiveness. On the other hand, ILC [7, 15, 148] is based on the notion that the performance of a system that executes the same task multiple times can be improved by learning from previous executions. Its objective is to improve performance by incorporating error information into the control for subsequent iterations. In doing so, high performance can be achieved with low transient tracking error despite large model uncertainty and repeating disturbances. Most of works relating to ILC are based on the repetitiveness of control process and considered for repetitive tracking tasks. As another learning-type control scheme, repetitive control (RC) [41, 47, 81, 85] is perhaps most similar to ILC except that RC is intended for continuous operation, whereas ILC is intended for discontinuous operation. In RC, the initial conditions are set to the final conditions of the previous trial. In ILC, the initial conditions are set to the same values on each trial. RC is often efficient to systems that operate in the whole time space. Neural networks (NN) [42, 122], or artificial neural networks to be more precise, represent

an emerging technology rooted in universal approximation (input-output mapping), the ability to learn from and adapt to their environment, and the ability to invoke weak assumptions about the underlying physical phenomena responsible for the generation of the input data. It performs useful computations through a process of "learning". NN is a good choice when non-parametric uncertainties are encountered.

Despite the existence of difference in learning process, it is a fact that the consistent target of all the learning-type control approaches is to achieve the asymptotic convergence property in tracking a given trajectory.

As two of the dominant components in learning-type control strategies, in this thesis, we put more effort to the design and analysis of adaptive control and iterative learning control. More introduction is given in the following for both of them.

1.1.1 Adaptive control

Adaptive Control is a systematic approach for automatic adjustment of the controllers in *real time*, in order to achieve or to maintain a desired level of performance of the control system when the parameters of the plant dynamic model are unknown and/or change in time.

Consider first the case when the parameters of the dynamic model of the plant to be controlled are unknown but constant (at least in a certain region of operation). In such case, while the structure of the controller will not depend in general upon the particular values of the plant model parameters, the correct tuning of the controller parameters cannot be done without the knowledge of those parametric values. Adaptive control techniques can provide an automatic tuning procedure in closed loop for the controller parameters. In such case, the effect of the adaptation vanishes as time increases. Changes of the operation conditions may require a re-start of the adaptation procedure.

Consider now the case when the parameters of the dynamic model of the plant change unpredictably in time. These situations occur either because the environmental conditions change or because we have considered simplified linear models for nonlinear systems. These situations may also occur simply because the parameters of the system are slowly time-varying. In order to achieve and to maintain an acceptable level of performance of the control system when large and unknown changes in model parameters occur, an adaptive control approach has to be considered. In such cases, the adaptation will operate most of the time and the non-vanishing adaptation fully characterizes this type of operation (sometimes called also continuous adaptation).

Extracting the constant feature and the time-varying feature of parameters simultaneously from the above two scenarios, there exists one special case in which the parameters of the dynamic model of the plant to be controlled are unknown but periodic, in the time domain or the space domain. These situations can be encountered in many rotational systems. Projection-based or least square-based algorithm can be adopted to adaptively learn their values in a pointwise way in each period. Considering this scenario as a direct extension from the constant unknown case, the linear growth condition and the linear structure in parameters are often assumed beforehand. Due to the fact that periodic variation of parameters could make the controller design much more complex, some useful techniques, e.g. the lifting technique in this thesis, are proposed to facilitate the AC design in the case.

An adaptive control system measures a certain performance index of the control system using the inputs, the states, the outputs and the known disturbances. From the comparison of the measured performance index and a set of given ones, the adaptation mechanism modifies the parameters of the adjustable controller and/or generates an

auxiliary control in order to maintain the performance index of the control system close to the set of given ones. Note that the control system under consideration is an adjustable dynamic system in the sense that its performance can be adjusted by modifying the parameters of the controller or the control signal. The above definition can be extended straightforwardly for "adaptive systems" in general. A conventional feedback control system will monitor the controlled variables under the effect of disturbances acting on them, but its performance will vary (it is not monitored) under the effect of parameter disturbances (the design is done assuming known and constant process parameters). An adaptive control system, which contains in addition to a feedback control with adjustable parameters a supplementary loop acting upon the adjustable parameters of the controller, will monitor the performance of the system in the presence of parameter disturbances. While the design of a conventional feedback control system is oriented firstly toward the elimination of the effect of disturbances upon the controlled variables, the design of adaptive control systems is oriented firstly toward the elimination of the effect of parameter disturbances upon the performance of the control system. An adaptive control system can be interpreted as a feedback system where the controlled variable is the performance index.

Many topics in adaptive control have been enthusiastically pursued over the past four decades. For instance, the effect of external disturbance, slow parameter variations, small discontinuities in parameters, sudden changes in reference inputs, unknown control directions, etc., have been investigated, methods for achieving robust controllers have been studied. Among the many questions that arise naturally in the context of adaptive systems, the most critical one concerns the stability of the overall adaptive system. It is only after the proof of stability for such a system was given in the late 1970s that

adaptive control became an accepted design methodology.

1.1.2 Iterative learning control

ILC is an approach for improving the transient performance of systems that operate repetitively over a fixed time interval. Although control theory provides numerous design tools for improving the response of a dynamic system, it is not always possible to achieve desired performance requirements, due to the presence of unmodeled dynamics or parametric uncertainties that are exhibited during actual system operation or to the lack of suitable design techniques. Thus, it is not easy to achieve perfect tracking using traditional control theories. ILC is a design tool that can be used to overcome the shortcomings of traditional controller design, especially for obtaining a desired transient response, for the special case when the system of interest operates repetitively. For such systems, ILC can often be used to achieve perfect tracking, even when the model is uncertain or unknown and we have no information about the system structure and nonlinearity.

ILC has been widely applied to mechanical systems such as robotics, electrical systems such as servomoters, chemical systems such as batch realtors, as well as aerodynamic systems, etc. ILC has been applied to both motion control and process control areas such as wafer process, batch reactor control, IC welding process, industrial robot control on assembly line, etc. Learning control system can enjoy the advantage of system repetition to improve the performance over the entire learning cycle.

Up to now, there are many approaches which can be employed to analyze ILC convergence property such as contraction mapping and energy function. Contraction mapping method is a systematic way of analyzing learning convergence. The global Lipschitz condition is a basic requirement which limits its extending to more general class of nonlinear

systems. Moreover, generally the contraction mapping design only cares the tracking convergence along learning horizon, while the system stability, which is an important factor in system control, is ignored. Therefore, energy function based ILC convergence analysis is widely applied for nonlinear systems. The most recent development of ILC focuses on several problems: ILC for non-repetitive task or plant, ILC for input nonlinearity, ILC for stochastic processes, and ILC for distributed parameter systems, etc.

1.2 Motivations

Adaptive control theory is one of the most well established theories in control area, and numerous results have been reported, e.g., [35, 60, 67, 90, 101]. By introducing a parametric adaptation mechanism, which essentially consists of a number of integrators, the adaptive control system is able to achieve asymptotic tracking convergence in the presence of parametric uncertainties. These uncertainties may be constant form, time-varying or state-varying. Most of previous efforts have been focused on the first two types. For instance, Ahn and Chen solved a time periodic adaptive friction compensation problem in [3]; Fidan et. al. discussed the adaptive control of a class of slowly time-varying systems with modelling uncertainties in [36]; Liu and Peng developed a method of time-varying disturbance compensation based on an observer in [78] and Xu introduced a time-periodic adaptive learning controller in [138]. In these references, the desired trajectories are always assumed to be time periodic or time dependent.

However, if control methods are always devised in time domain, then the information available through the underlying nature of the system will possibly not be fully captured and utilized, such as state-periodicity of system uncertainties. As a result, the control problem can not be solved properly. Thus, to discuss state-dependent system in state

(e.g., position or speed) domain sometimes is more reasonable and meaningful.

In practice, the state-dependent external uncertainties (e.g., position dependent or velocity dependent uncertainties) exist in various engineering problems. For example, in [22] and [168], the engine crankshaft speed pulsation was expressed as Fourier series expansion as a function of position; in [43], the external disturbance of the satellite was modelled as a function of the position; in [167], for a Permanent Magnet Synchronous Motor (PMSM) system the uncertainty of the observer-based robust adaptive control was also related to rotor position. Moreover, [165] proposed an angle-based control method to rotate the pendulum and to stabilize the base link, which is designed by the state-dependent Riccati equation based on zero-dynamics of the pendulum; and [13] discussed how to handle state-dependent nonlinear tunnel flows in short-term hydropower scheduling. More examples can be seen from appliances of alternating current, investigation of nonlinear frictions, vehicle systems and other rotary machine systems.

In the rotary machine systems mentioned above, the existent uncertainties are usually periodic in state domain but not in time domain. Relatively, few research efforts have been devoted to these state-dependent problems from the view of generality. Among the literature, [24] and [26] were devoted to the problem of rejecting oscillatory position-dependent unknown disturbance (eccentricity) with a sinusoidal form, where they formulate and globally solve the adaptive cancelation problem in the spatial domain coordinates. In [40], the speed of the servo-motor was controlled with a position-dependent unknown disturbance using iterative learning control. More generally, [4] extended this sort of problem to a general wave form, which portraits the unknown position-dependent periodic disturbance.

Basing on the known results for time-dependent and state-dependent parametric

uncertainties, in order to deal with spatial periodic control for rotating machine systems, a fundamental task is how to control the plant with highly nonlinear components, such as local Lipschitzian and continuous functions, to track a nonlinear reference model, either periodic or even non-periodic. In the first part of the thesis, our attention is paid to this issue.

Moreover, as we stated before, periodic variations are invariant under a shift by one or more periods, and they are often a consequence of some rotational motion at constant speed and encountered in many real systems such as electrical motors, generators, helicopter blades and satellites [27, 28, 31, 61, 64, 95, 127, 150]. As in the case of linear periodic systems, many results have been achieved to deal with their adaptive control, robustness and identification [54, 91, 120]. Recently, discrete-time periodic adaptive control (PAC) has been proposed and the underlying idea of PAC is to update parameters in the same instance of two consecutive periods [1, 45]. Due to the time-varying nature, it would be very difficult, if not impossible, to design appropriate periodic adaptive controllers for more general scenarios such as plants with unknown control directions, plants in parametric-strict-feedback form, plants with nonlinear parameterization, plants not satisfying any growth conditions, etc. For instance, the periodic updating law [1] is not extendable to the plants without any growth conditions in nonlinearities as was achieved in [65], due to the fundamental difference between classical adaptive control and PAC: the former is updated between two consecutive time instances whereas the latter is updated between two consecutive period which incurs a delay of one period.

Actually, many effective adaptive control methods have been developed for discrete-time systems with time-invariant parametric uncertainties, such as [166] for parametric-strict-feedback form, [39, 74] for unknown control direction, [65] for plants without any

growth conditions in nonlinearities. It would be highly desirable if we can apply these well established adaptive control methods to plants with periodic parameters. To achieve this objective, we adopt a lifting technique to convert periodic parameters into an augmented vector of time-invariant parameters, in the sequel all existing adaptive control methods can be applied. Although a simple lifting idea is proposed and applied over here, many open problems to periodic parametric systems can be solved clearly.

AC is an efficient method to deal with systems with parametric repetitiveness, and the ultimate tracking convergence is derived in time space. Nevertheless, many systems with other kinds of repetitiveness can not be addressed by this technique. Next, we state some motivations relevant to systems with repetitiveness that can be solved by ILC methodology.

ILC was firstly proposed by Arimoto et al. [8]. After that, many research work has been carried out in this area and a lot of systematic approaches have been developed for a large variety of linear or nonlinear systems to deal with repeated tracking control problems or periodic disturbance rejection problems. ILC has been proposed and developed as a kind of contraction mapping approach to achieve perfect tracking under the repeatable control environment which implies a repeated exosystem in a finite time interval with a strict initial reseting condition, [9], [114], [88], [132], etc.

Recently, new ILC approaches based on Lyapunov function technology [96], [97] and Composite Energy Function (CEF) [133], [134] have been developed to complement the contraction mapping based ILC. For instance, by means of CEF based ILC, we can extend the system nonlinearities from global Lipschitz continuous to non-global Lipschitz continuous [133], extend target trajectories from uniform to non-uniform ones [135], remove the requirement on the strict initial resetting conditions [136], deal with time varying and

norm bounded system uncertainties [134], and incorporate nonlinear optimality [137], etc.

Different from many known applications of ILC method, there are some circumstances in which one can not control an object any more after an initial command signal is given to it. For example, in basket ball shooting, once the ball has left the player's hand, it is impossible to modify the flight trajectory of the ball by means of any sensory feedback. Actions of this type are called *ballistic* [23]. The ballistic characteristic is the main feature of many sports items, such as archery, bowling, dart, or any ball games. Ballistic control is also widely encountered in military training and practice, such as projectile, shooting, etc. The well known instantaneous feedback, or on-line feedback, is not applicable to this class of control tasks while it is still necessary to work out beforehand the desired command needed in order to achieve the goal. Usually, the initial command signal is characterized by the initial state of system. Note that for these circumstances, initial state is just the adjustable system input. Thus, such an Initial State Learning (ISL) problem is fully different from the discussed ISL problems before, such as in [29], where input term exists instantaneously in the discussed dynamic system and the initial state is only an initial condition of system operation.

However, it is often difficult to calculate the proper command signal sequences in advance. Such a prior calculation requires the complete knowledge of the entire process involved in the control, such as the object dynamic model, parameters, interactions with environment, the actuation mechanism, precise sensory information, etc.. In the real world it is hard to have the perfect knowledge. What we do in practice is to build up an internal model via repeated learning. This internal model will generate the appropriate command signals directly for a given task, instead of deriving a perfect model for the

task and conducting a model based calculation. Hence it is meaningful to develop a powerful scheme which can efficiently characterize the learning process of this class of ISL problems.

Consider another typical example: the process for a train slipping into a station, where the friction of rail is an uncertainty. Define our control objective firstly as to implement the behaviors like traction and brake as least as possible to control spatial states of system, i.e., make the train go across the desired position with a desired speed. This is meaningful since it can depress the oil consumption and reduce the damage towards train and rail. If the train can be controlled to slip freely from an appropriate position with an appropriate initial speed, then the aim can be attained. Obviously, this problem also belongs to ISL category.

The ultimate aim of ISL is obviously to realize the final state regulation. We know that final state control problems have been widely explored, such as in [15, 147, 164]. However the control signals used in these methods are continuously applied throughout the operation period. Two-point boundary-value problems also consider initial and final state relations [50], but the solutions are numerically solvable only when the dynamics is completely known. As the first application of ILC, we formulate this problem as a motion control problem and focus on the learning convergence of spatial initial states in planar autonomous systems. Our analysis reveals that ILC is an efficient method to deal with the sort of control task and all the learning behaviors can be illustrated very well.

In recent decades, nonlinear system control with input uncertainties has received a great deal of attention, since input uncertainties are quite common phenomenon in engineering applications. Examples of input uncertainties include saturation, deadzone, hysteresis and so on. The existence of these input uncertainties may severely deteriorate

the control performance or cause oscillations, even lead to system instability [118, 143–146].

As one of the well-known control techniques, ILC has demonstrated its ability to deal with this sort of issue when the control environment is repeatable. In [143, 144], an ILC scheme is designed for a class of nonlinear uncertain systems with input saturation. The analysis of convergence in the iteration domain is based on composite energy function, which consists of both input and state information along the time and iteration axes. More early, [32] extended the ILC method of Arimoto *et al.* [8] for MIMO system to the scenario that each component of input is bounded and rate-limited. Using discrete-time Lambda norm, monotonic convergence was derived in norm for the input error sequence. In the above three works, a fundamental fact is used: the saturation operator for control input will not enlarge its error to the desired input that lies in this interval. Besides these, we can see that the tracking problem of linear systems with input constraints was formulated as a constrained convex optimization problem, namely a linearly constrained quadratic program, and an interior point algorithm, specifically the barrier method, was adopted to solve the proposed ILC problem in [87]; the robust stability criteria of a single-input-single-output (SISO) ILC system with friction and input saturation process, using frequency domain methods and 2-D system theory, was investigated in [51]. It is proved in [145, 146] that ILC methodology remains effective for systems having an input deadzone that could be nonlinear, unknown and state-dependent. Despite the presence of the input deadzone, the simplest ILC scheme retains its ability to achieve satisfactory performance. Recently, as can be seen in [117], ILC is further considered with a general input uncertainty which may take saturation or deadzone form, where a dual iterative learning loop is constructed to learn both the unknown nominal dynamics and the input

static mapping.

So far, however, much less work has been done to dynamic systems with hysteretic input uncertainty, although ILC design for hysteresis system has been frequently discussed [52, 53, 72, 79]. The difficulty in proving convergence of ILC algorithms for hysteretic systems arises due to two reasons: (i) branching effects and (ii) nonlinearity of each branch [21]. The latter issue can be addressed by standard ILC methods. For example, the convergence of ILC on a single branch was shown in [52], in which the hysteresis nonlinearity was modeled as a single branch (using a polynomial). Alternatively, a functional approach was proposed for systems that satisfy the incrementally strictly increasing operator (ISIO) property [125]; however, the branching effect in hysteresis results in loss of the ISIO property [77]. The reason branching causes problems in proving convergence is because branching prevents the ILC algorithm from predicting the direction in which the input needs to be changed based on a measured output error. In [71], this problem has been addressed by constructing the monotonic property between input and output for a Preisach model.

Hysteresis is a very complex phenomenon and there exist many hysteresis models in literature, e.g., the Bouc-Wen model, Duhem model, the Jiles-Atherton model, the Prandtl-Ishlinskii model, and the Preisach model. A fact is that almost all the previous ILC design schemes are focused on the Preisach model, if hysteresis is addressed. However, as another typical class, the Bouc-Wen model for smooth hysteresis has received an increasing interest due to its capability to capture in an analytical form a range of shapes of hysteretic cycles which match the behavior of a wide class of hysteretic systems [56–58, 92, 105, 112]. In particular, it has been used experimentally to model piezoelectric elements, magnetorheological dampers, wood joints and base isolation de-

vices for buildings. The obtained models have been used either to predict the behavior of the physical hysteretic element or for control purposes.

As the second application of ILC strategy, we address the ILC problem for a simple scalar nonlinear dynamic system with a hysteresis input uncertainty, which takes the structure of the Bouc-Wen model. By analyzing the input-output monotonicity of the hysteresis part in plant and considering a dual loop ILC structure, the output tracking convergence can be derived by a rigorous Lyapunov function based analysis.

While for processes described by ordinary differential equations (ODEs) many control schemes have been proposed, fewer control schemes have been developed for processes described by partial differential equations (PDEs). A major portion of established PDE control schemes focus on the use of distributed actuation, namely, the control action depends on the spatial coordinates. However, in many important industrial processes the control actuation is achieved through the boundary of the process, such as the case of chemical and biochemical reactors where the manipulated input is the fluid velocity at the feed of the process [37, 63].

In [107]- [110], the boundary control of PDEs with adaptive control methodology is extended to cope with either stable or unstable PDEs. These works are built upon explicitly parameterized control formulae to avoid solving Riccati or Bezout equations at each time step. Backstepping is also adopted to solve the problem of stabilization of some PDEs by using boundary control in [111] [69]. In practice, however, simple controllers such as PI or PID compensators are most widely used by process engineers in the chemical and biochemical industry, owing to many reasons such as implementability, the long history of proven operation and robustness, and the fact that these simple controllers are well understood by industrial practitioners.

A major difficulty in PDE control is how to optimally tune the controller gains. When process uncertainties are present, it is almost impossible to find the values or bounds of the controller gains such that the closed-loop performance can be guaranteed for the PDE processes, as can be seen from [63].

As the third application of ILC strategy, we assume that the considered PDE procuress is strictly repeatable, which is one of the main features in certain types of real process control including industrial chemical [34] and biochemical reactors [37], and then develop the ILC for a class of single-input single-output (SISO) nonlinear PDE processes with boundary control and containing unknown parameters affecting the interior of the domain. The control objective is to iteratively tune the velocity boundary condition on one side such that the boundary output on the other side can be regulated to a desired level.

Assuming the repetitiveness of PID control process, we are now at the position of considering the optimal tuning of PID parameters using iterative learning approach. Among all the known controllers, the proportional-integral-derivative (PID) controllers are always the first choice for industrial control processes owing to the simple structure, robust performance, and balanced control functionality under a wide range of operating conditions [33, 62]. Although being widely used in industry, tuning PID parameters (gains) remains a challenging issue and directly determines the effectiveness of PID control [33,66]. To address the PID design issue, much effort has been invested in developing systematic auto-tuning methods. These methods can be divided into three categories, where the classification is based on the availability of a process model and model type, (i) model free methods; (ii) non-parametric model methods and (iii) parametric model methods.

In model free methods, no model or any particular points of the process are identified. The non-parametric model methods use partial modelling information, usually including the steady state model and critical frequency points. These methods are more suitable for online use and applied without the need for extensive priori plant information [62]. Relay feedback tuning method [12,124] is a representative method of the second category. The parametric model methods require a linear model of the process – either transfer function matrix or state space model. To obtain such a model, standard off-line or on-line identification methods are often employed to acquire the model data. Thus parametric model methods are more suitable for off-line PID tuning [10].

It should be noted that in many industrial control problems such as in process industry, the process is stable in a wide operation range under closed-loop PID, and the major concern for a PID tuning is the transient behaviors either in the time domain, such as peak overshoot, rise time, settling time, or in the frequency domain such as bandwidth, damping ratio and undamped natural frequency. From the control engineering point of view, it is one of the most challenges to directly address the transient performance, in comparison with the stability issues, by means of tuning control parameters. Even for a lower order LTI process under PID, the transient performance indices such as overshoot could be highly nonlinear in PID parameters and an analytical inverse mapping from overshoot to PID parameters may not exist. In other words, from the control specification on overshoot we are unable to decide the PID parameters analytically. The first objective we want to realize is to link these transient specifications with PID parameters and give a systematic tuning method.

In practice, when the process model is partially unknown, it would be difficult to calculate the PID parameters even if the nonlinear mapping between the transient spec-

ifications and PID parameters can be derived. In existing PID tuning methods, whether model free or model based, test signals will have to be injected into the process in order to find certain relevant information for controller parameter setting. This testing process may however be unacceptable in many real-time control tasks. On the other hand, many control tasks are carried out repeatedly, such as in batch processors. Thus, we want to further explore the possibility of fully utilizing the task repetitiveness property, consequently provide a learning approach to improve PID controllers through the iteratively parameter tuning when the transient behavior is of the main concern.

In most learning algorithms including neural learning and iterative learning, the process Jacobian or gradient plays the key role by providing the greatest descending direction for the learning mechanism to update inputs. The convergence property of these learning algorithms is solely dependent on the availability of the current information on the gradient. The gradient between the transient control specifications and PID parameters, however, may not be available if the plant model is unknown or partially unknown. Further, the gradient is a function of PID parameters, thus the magnitude and even the sign may vary. The most difficult scenario is when we do not know the sign changes *a priori*. In such circumstances, traditional learning algorithms cannot achieve learning convergence. Lastly, we hope to extend the iterative learning approach to deal with the unknown gradient problem for PID parameter tuning.

Another issue is concerned with the redundancy in PID parameter tuning when only one or two transient specifications are required. In order to fully utilize the extra degrees of freedom, the most common approach is to introduce an objective function and optimize the PID parameters accordingly. This traditional approach is however not directly applicable because of the unknown process model, and in particular the unknown varying

gradient. A key to the solution of this problem is still iterative learning. An objective function, which is accessible, is chosen as the first step for PID parameter optimization. Since the goal is to minimize the objective function, the control inputs will be updated along the greatest descending direction, namely the gradient, of the objective function. In other words, the PID parameters are chosen to directly reduce the objective function, and the objective function is treated as the process output and used to update the PID parameters. When the gradient is varying and unknown, extra learning trials can be conducted to search the best descending direction.

1.3 Objectives and Contributions

In this thesis, the research is focused on developing several learning-type control approaches for nonlinear dynamic systems with repetitiveness. The main contributions lie in the following aspects: AC design for systems with periodic repetitiveness in parametric form, ILC design for systems with process repetitiveness, and more iterative tuning or identification of parameters for systems with certain repetitiveness. The contributions of the thesis are summarized in Table 1.1. In details, the contributions of this thesis are as follows:

(1). In Chapter 2, a new spatial periodic adaptive control approach is proposed to deal with nonlinear rotary machine systems with a class of state-varying parametric uncertainties, which are in an unknown compact set, periodic, non-vanishing, and the only prior knowledge is the periodicity. In this process, we make full use of the system information regarding uncertainties and nonlinearities w.r.t. spatial states. For instance, since any periodic function is cycle-invariant, we design our control actions with this invariance. Moreover, we also focus on the relationship between the systems with state-dependent

<table>
<tr><th colspan="2">Dynamic System</th><th colspan="2">Repetitiveness</th><th>Control Method</th><th>Performance</th></tr>
<tr><td colspan="2">Linear system</td><td colspan="2">Control task, System uncertainties</td><td>PID with ILT</td><td>Asym. conv. in iteration domain</td></tr>
<tr><td colspan="2">Semilinear PDE system</td><td colspan="2">Control task, System uncertainties</td><td>ILC based on CM</td><td>Mono. conv. in iteration domain</td></tr>
<tr><td rowspan="4">Nonlinear system</td><td rowspan="3">Continuous time</td><td>Periodicity of para. uncertainties</td><td>Spatial domain</td><td>SPAC based on LKF</td><td>$\|\cdot\|_L$ convergence in spatial domain</td></tr>
<tr><td rowspan="2">Control task, Para. uncertainties</td><td>Spatial domain</td><td>ILC based on CM</td><td>Mono. conv. in iteration domain</td></tr>
<tr><td>Time domain</td><td>ILC based on CM and CEF</td><td>Uniformly bounded</td></tr>
<tr><td>Discrete time</td><td>Periodicity of para. uncertainties</td><td>Time domain</td><td>AC with the lifting technique</td><td>Asym. conv. in time domain</td></tr>
</table>

Table 1.1: The contribution of the thesis. AC: adaptive control, ILC: iterative learning control, ILT: iterative learning tuning, PAC: periodic adaptive control, SPAC: spatial periodic adaptive control, CM: contraction mapping, CEF: composite energy function, LKF: Lyapunov-Krasovskii functional, Asym. conv.: asymptotical convergence, Mono. conv.:monotonic convergence, Para.: Parametric, $\|\cdot\|_L \triangleq \sup_{s\geq L}\int_{s-L}^{s}\|\cdot\|^2 d\tau$.

uncertainties and the systems with time-dependent uncertainties. On the one hand, by making full use of the system information regarding uncertainties and nonlinearities in state space, the tracking problem becomes possible. On the other hand, to solve such tracking problem pushes us to investigate the relationship between the systems with state-dependent uncertainties and the systems with time-dependent uncertainties and try to apply them. As a result, by applying these known achievements, our difficulty is overcome properly in spatial control for the plant with highly nonlinear components.

(2) In Chapter 3, a concise discrete-time adaptive control approach suitable for nonlinear systems with periodic parametric uncertainties is proposed, based on the lifting technique. By using such a technique, the periodic parameters are converted into an augmented constant parametric vector, and then the well-established discrete-time adaptive control schemes can be easily applied to various control problems with periodic parameters, such as plants with unknown control directions, plants with unknown control gains, and plants in parametric-strict-feedback form, plants that do not meet the linear growth condition, etc. Another major advantage of the new adaptive control is the ability to adaptively update all parameters in parallel, hence expedite the adaption speed.

(3). An initial state ILC approach is proposed for final state control of motion systems in Chapter 4. ILC is applied to learn the desired initial states in the presence of system uncertainties. Four cases are considered where the initial position or speed are manipulated variables and final displacement or speed are controlled variables. In these cases, the motion system could have discontinuous damping or discontinuous frictions but Lipschitzian in position. By duality, we further explore other four cases if the motion system is Lipschitz continuous in speed. Since the control task is specified spatially in states, a state transformation is introduced such that the final state control problems are

formulated in the phase plane to facilitate spatial ILC design and analysis.

(4). In Chapter 5, a dual-loop ILC scheme is designed for a class of nonlinear systems with hysteresis input uncertainty. The two ILC loops are applied to the nominal part and the hysteresis part respectively, to learn their unknown dynamics. Based on the convergence analysis for each single loop, a composite energy function method is then adopted to prove the learning convergence of the dual-loop system in iteration domain, where the input-output monotonicity in each branch of hysteresis is a key point. When the strict input-output monotonicity is violated in the hysteretic loop, the ILC law is revised by adding a forgetting factor and incorporating a time-varying learning gain, and then ensure the corresponding ILC operator to be contractible. Using the Banach fixed-point theorem, we show that the output tracking error of the inner ILC loop and then the dual ILC loop can enter and remain ultimately in a small neighborhood of zero.

(5). In Chapter 6, the ILC scheme is developed for a class of nonlinear PDE processes with boundary control and containing uncertainties affecting the interior of the domain. The control objective is to iteratively tune the velocity boundary condition on one side such that the boundary output on the other side can be regulated to a desired level. Under certain practical properties such as physical input-output monotonicity, process stability and repeatability, the problem is simplified as an output regulation problem in spatial domain only. By means of rigorous analysis, the learning convergence is achieved under repeatable process environment.

(6). It is proposed in Chapter 7 that an iterative learning tuning method – an optimal tuning method for PID parameters by means of iterative learning. PID parameters are updated whenever the same control task is repeated. The first novel property of the new

tuning method is that the time domain performance or requirements can be incorporated directly into the objective function to be minimized. The second novel property is that the optimal tuning does not require as much the plant model knowledge as other PID tuning methods. The new tuning method is essentially applicable to any plants that are stabilizable by PID controllers. The third novel property is that any existing PID auto-tuning methods can be used to provide the initial setting of PID parameters, and the iterative learning process guarantees that a better PID controller can be achieved. The fourth novel property is that the iterative learning of PID parameters can be applied straightforward to discrete-time or sampled-data systems, in contrast to existing PID auto-tuning methods which are dedicated to continuous-time plants. In this chapter, we further exploit efficient searching methods for the optimal tuning of PID parameters. Through theoretical analysis, comprehensive investigations on benchmarking examples, and real-time experiments on the level control of a coupled-tank system, the effectiveness of the proposed method is validated.

Chapter 2

Spatial Periodic Adaptive Control for Rotary Machine Systems

2.1 Introduction

Rotary machine systems are widely used in industries. Two representative classes of rotary machines are the electrical motor drives and vehicle engines. Electrical motor drives, including DC servos, induction motors, permanent magnetic synchronous motors (PMSM), switched reluctance motors, are typical rotary machine systems that convert electrical energy into mechanical energy. We can find numerous applications of such rotary mechanisms as hard disk drives, robot manipulators, conveyors, etc. Engine systems in vehicles and aircrafts are another class of rotary machine systems that convert fuel energy into rotational work.

A fundamental property of any rotary machine systems is the spatial periodicity in terms of angular displacement, that is, the angular displacement will come back to the same angular position after rotating certain degrees. This spatial periodicity is independent of the speed of rotational machines. On the other hand, a large class of system uncertainties in rotary machines are related to the angular position. In [22]

and [168], the unknown engine crankshaft speed pulsation is expressed as Fourier series of the angular position. In [43], the external disturbance of the satellite is modelled as a function of the position. In [13], the position-dependent nonlinear tunnel flows in short-term hydropower scheduling was discussed. In general, this class of uncertainties can be modeled as either periodic unknown parameters or periodic unknown disturbances with respect to (w.r.t.) the angular displacement.

Adaptive and learning control approaches were proposed to deal with the position or state-dependent periodic uncertainties. In [40], learning control was used when the position-dependent disturbance torque is presented in servo motor under velocity control. In [26], adaptive compensation was developed to reject oscillatory position-dependent disturbance in a sinusoidal form without knowing the amplitude and frequency. In [4] and [5], periodic adaptation was developed to handle the unknown position-dependent periodic disturbance. In [104], a feedback linearization was developed for temporal-spatial conversion where rotational hydraulic drive was under consideration.

In this work, we extend the spatial periodic adaptive control (SPAC) approach to more generic classes of control problems with periodic parameters or periodic disturbances. In practical rotary machine systems, these unknown parameters or disturbances are either smooth functions or continuous functions of the angular displacement, hence can be approximated by Fourier series or other function approximation methods. Since it is impossible to implement an infinite series in a practical controller, we introduce a delay type periodic adaptation law which consists of only two terms but of infinite dimensions.

Another extension is to high order rotary systems with the tool of feedback linearization. The extension of the SPAC to high order systems is not straightforward, even if

the original high order system is in the canonical form in the time domain. In SPAC, the system dynamics is converted from the time domain to the spatial domain. The objective of the temporal-spatial conversion is to capture and fully utilize the spatial periodic characteristic of the process uncertainties, so that the controller and the spatial periodic adaptation can be designed in the spatial domain. The extra difficulty arises when the temporal-spatial conversion is carried out. A canonical dynamics in the time domain is no longer canonical in the spatial domain. In this chapter, to address this issue, a feedback linearization is proposed such that both the process dynamics and the reference model can be strictly linearized into the canonical form.

The third extension is to high order systems with multiple periods or pseudo-periods. In the presence of multiple periods which are rational numbers, the periodic adaptation can be carried out according to the lowest common multiple. However, the use of the common period will make the periodic adaptation inefficient. For example, suppose a period is 3 and another is 100. The lowest common multiple is 300. As a result, the periodic adaptation for the period of 3 has been delayed by 100 cycles. If possible, the periodic adaptation should be conducted according to individual periods. In pseudo-periodic circumstances where periods are mixed with rational and irrational numbers such as 3 and $\sqrt{3}$, or irrational numbers such as $\sqrt{3}$ and π, there does not exist a common period. To address this issue, we develop a SPAC which can conduct periodic adaptation in parallel for all parameters with different periods.

In order to facilitate the property analysis in SPAC especially for pseudo-periodic circumstances, we introduce a Lyapunov-Krasovskii functional (LKF) as a generic tool. By virtue of the LKF, we can show the asymptotic convergence of the speed tracking error, and the boundedness of the system states and the control input.

The chapter is organized as follows. In Section 2.2, some preliminaries with regard to convergence properties are provided for subsequent sections. In Section 2.3, we focus on SPAC by considering high order systems with the feedback linearization. In Section 2.4, SPAC is further extended to systems with pseudo-periodic parameters. In Section 2.5, two illustrative examples are provided.

2.2 Preliminaries

Definition 2.1 *When analyzing a vector valued function* $\mathbf{f}(s)$*, an important quantity is the integral over an interval of length* L*, namely* $\int_{s-L}^{s} \|f(\tau)\|^2 d\tau$*, where* $\|\cdot\|$ *is the 2-norm.* $\mathbf{f}(s)$ *is* L*-bounded if* $\sup_{s\geq L} \int_{s-L}^{s} \|\mathbf{f}(\tau)\|^2 d\tau$ *is finite, and that* $\mathbf{f}(s)$ *is* L*-convergent if* $\lim_{s\to\infty} \sup_{s\geq L} \int_{s-L}^{s} \|\mathbf{f}(\tau)\|^2 d\tau = 0$.

Definition 2.2 *A matrix-valued function* $\Gamma(s, L) = diag\{\gamma_1(s, L), \cdots, \gamma_m(s, L)\}$ *is defined in the interval* $[0, \infty)$*, satisfying*

$$\Gamma(s, L) = \begin{cases} 0, & s = 0, \\ \mathcal{A}(s), & 0 \leq s < L, \\ \mathcal{B}, & s \geq L, \end{cases} \tag{2.1}$$

where $\mathcal{A} = diag\{\alpha_1(s), \cdots, \alpha_m(s)\}$ *and* $\mathcal{B} = diag\{\beta_1, \cdots, \beta_m\}$ *are diagonal matrices,* $\alpha_i(s)$ *is a strictly increasing function for* $s \in [0, L]$ *with* $\alpha_i(0) = 0$*,* $\alpha_i(L) = \beta_i$*, and* $\beta_i > 0$ *is a constant.*

To facilitate the convergence analysis of SPAC, the following Lyapunov-Krasovskii functional (LKF) is adopted

$$V(s, \mathbf{e}, \boldsymbol{\phi}) = \frac{1}{2}\mathbf{e}^T\mathbf{e} + \frac{1}{2}\int_{\max\{0,s-L\}}^{s} \boldsymbol{\phi}^T(\tau)\mathcal{B}^{-1}\boldsymbol{\phi}(\tau)d\tau, \tag{2.2}$$

where $\mathbf{e} \in \mathbf{R}^n$, $\boldsymbol{\phi} \in \mathbf{R}^m$. For simplicity denote $V(s, \mathbf{e}, \boldsymbol{\phi})$ by $V(s)$ in subsequent context.

The L-convergence property associated with the LKF in (2.2) can be derived as shown the *Proposition* 2.1. Define a differential operator $\nabla = d/ds$ where s is a coordinate.

Proposition 2.1 *For the LKF defined in (2.2), if* $\nabla V \leq -g(\mathbf{e})$ *for* $s \in [L, \infty)$ *where* $g(\mathbf{e}) \geq 0 \ \forall \mathbf{e} \in \mathbf{R}^n$, *and the LKF is finite at* $s = L$ *for any constant* $L > 0$, *that is,* $V(L) < \infty$, *then*

$$\lim_{s\to\infty} \int_{s-L}^{s} g(\mathbf{e}) d\tau = 0. \tag{2.3}$$

Proof: Suppose that $\lim_{s\to\infty} \int_{s-L}^{s} g(\mathbf{e}) d\tau \neq 0$. There exist an $\varepsilon > 0$, $s_m \geq L$, a sequence $s_i \to \infty$ with $i = 1, 2, \cdots$ and $s_{i+1} \geq s_i + L$ such that

$$\int_{s_i-L}^{s_i} g(\mathbf{e}) d\tau > \varepsilon$$

when $s_i > s_m$. Hence,

$$\begin{aligned} \lim_{s\to\infty} V(s) &= V(L) + \lim_{s\to\infty} \int_{L}^{s} \nabla V(\tau) d\tau \leq V(L) - \lim_{s\to\infty} \int_{L}^{s} g(\mathbf{e}) d\tau \\ &\leq V(L) - \lim_{i\to\infty} \sum_{j=1}^{i} \int_{s_j-L}^{s_j} g(\mathbf{e}) d\tau \leq V(L) - \varepsilon \cdot \lim_{i\to\infty} i. \end{aligned}$$

Since $V(L)$ is finite, the above relationship implies $\lim_{s\to\infty} V(s) = -\infty$, a contradiction to the non-negativeness property of $V(s)$. Thus the L-convergence property (2.3) must hold. ■

Next we derive the boundedness of the LKF in the interval $[0, L]$ under certain conditions. Denote $\boldsymbol{\phi}(s) = \mathbf{a}(s) - \hat{\mathbf{a}}(s)$, where $\mathbf{a}, \hat{\mathbf{a}} \in \mathbf{R}^m$ and $\mathbf{a}$ has a vector valued upper bound $\bar{\mathbf{a}}$.

Proposition 2.2 *For* $s \in [0, L]$, $V(s)$ *is bounded if the following equality holds*

$$\nabla V(s) = -\lambda \|\mathbf{e}\|^2 + \boldsymbol{\phi}^T \mathcal{A}^{-1}(s) \hat{\mathbf{a}} + \frac{1}{2} \boldsymbol{\phi}^T \mathcal{B}^{-1} \boldsymbol{\phi}. \tag{2.4}$$

Proof: (2.4) can be rewritten as

$$\begin{aligned}\nabla V(s) &= -\lambda\|\mathbf{e}\|^2 - \boldsymbol{\phi}^T\mathcal{A}^{-1}(\mathbf{a}-\hat{\mathbf{a}}-\mathbf{a}) + \frac{1}{2}\boldsymbol{\phi}^T\mathcal{B}^{-1}\boldsymbol{\phi} \\ &= -\lambda\|\mathbf{e}\|^2 - \boldsymbol{\phi}^T(\mathcal{A}^{-1}-\mathcal{B}^{-1}/2)\boldsymbol{\phi} + \boldsymbol{\phi}^T\mathcal{A}^{-1}\mathbf{a}, \end{aligned} \tag{2.5}$$

where $\mathcal{A}^{-1}-\mathcal{B}^{-1}/2 > 0$ because $\alpha_i(s)$ is strictly increasing with the upper limit β_i. Using Young's inequality, we have for any $C = diag\{c_1, \cdots, c_m\} > 0$

$$\boldsymbol{\phi}^T\mathcal{A}^{-1}\mathbf{a} \le \boldsymbol{\phi}^T C\mathcal{A}^{-1}\boldsymbol{\phi} + \frac{1}{4}\mathbf{a}^T C\mathcal{A}^{-1}\mathbf{a}. \tag{2.6}$$

Choose C such that $\mathcal{A}^{-1} - \mathcal{B}^{-1}/2 - C > 0$. Substituting (2.6) into (2.5) yields

$$\nabla V(s) = -\lambda\|\mathbf{e}\|^2 - \boldsymbol{\phi}^T(\mathcal{A}^{-1}-\mathcal{B}^{-1}/2-C)\boldsymbol{\phi} + \frac{1}{4}\mathbf{a}^T C\mathcal{A}^{-1}\mathbf{a}.$$

Accordingly $\nabla V(s)$ for $s \in [0, L]$ is negative definite outside the region

$$\left\{(\|\mathbf{e}\|, \|\boldsymbol{\phi}\|) \in \mathbf{R}^2 \quad : \quad \lambda\|\mathbf{e}\|^2 + \lambda_1\|\boldsymbol{\phi}\|^2 \le \lambda_2\|\bar{\mathbf{a}}\|^2\right\}, \tag{2.7}$$

where $\lambda_1 > 0$ is the minimum eigenvalue of the matrix $\mathcal{A}^{-1} - \mathcal{B}^{-1}/2 - C$, and $\lambda_2 < \infty$ is the maximum eigenvalue of the matrix $C\mathcal{A}^{-1}/4$. From (2.7) we conclude the boundedness of $V(s)$ in the interval $[0, L]$. ■

Now investigate the relationship between the spatial and temporal coordinates. Denote t the time axis, s the angular displacement of rotary systems, and $x_1 = ds/dt$ is the angular speed. The spatial differentiator, or the ∇-operator, is defined below and linked to the temporal differentiator

$$\nabla = \frac{d}{ds} = \frac{d}{dt}\left(\frac{ds}{dt}\right)^{-1} = \frac{1}{x_1}\frac{d}{dt}. \tag{2.8}$$

Let us further explore the relationship between the temporal coordinate t and spatial coordinate s, so as to facilitate the conversion between t and s. From $ds = x_1 dt$ we have

$$s = \int_0^t x_1(\tau)d\tau \triangleq f(t). \tag{2.9}$$

When the angular speed $x_1 > 0$, s is a strictly increasing function of t, hence the relationship between t and s is bijective. The function $s = f(t)$ is analytic and the inverse function $t = f^{-1}(s)$ exists globally. Therefore a variable, $x(t)$, which is a temporal function, can also be expressed as a spatial function $x(f^{-1}(s))$.

Throughout this chapter, we make the following assumption.

Assumption 2.1 *The rotary system under consideration is evolving in one direction, and the speed of the rotary system is strictly above zero, that is* $x_1 > 0 \ \forall t$.

Remark 2.1 *For rotary systems, rotating direction and rotating speed are most important characteristics. In many circumstances, speed regulation is often the aim of control. Many motion systems run around an operating point specified by a constant speed, such as the electrical drives used in escalators, engines used in vehicle cruise. When the rotational motion speed occasionally crosses zero but most time run at a nonzero speed, we can simply switch to another controller at near zero speed, and back to the periodic adaptive control when the speed is near the operating point. If the rotational motion speed frequently crosses zero such as in car parking, the angular displacement of the rotational mechanism is unlikely to show any cyclic behavior. In such circumstances, the periodic adaptive control is not a suitable control method.*

To facilitate the analysis of SPAC, the algebraic relationship

$$(\mathbf{a}-\mathbf{b})^T\mathcal{B}(\mathbf{a}-\mathbf{b}) - (\mathbf{a}-\mathbf{c})^T\mathcal{B}(\mathbf{a}-\mathbf{c}) = (\mathbf{c}-\mathbf{b})^T\mathcal{B}[2(\mathbf{a}-\mathbf{b}) + (\mathbf{b}-\mathbf{c})], \qquad (2.10)$$

is introduced, where $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ are vectors with same dimensions and $\mathcal{B}$ is the diagonal gain matrix.

For simplicity, we omit all the arguments from a function where no confusion arises, e.g. denote $f(\cdot,\cdot)$ by f.

2.3 SPAC for High Order Systems with Periodic Parameters

Consider the canonical system

$$\begin{cases} \dfrac{ds}{dt} &= x_1, \\ \dfrac{dx_i}{dt} &= x_{i+1}, \qquad i = 1, 2, \cdots, n-1, \\ \dfrac{dx_n}{dt} &= \mathbf{a}^T(s)\boldsymbol{\zeta}^0(\mathbf{x}) + b(s)u, \end{cases} \tag{2.11}$$

where $\mathbf{x} = [s, x_1, \cdots, x_n]^T$. $\mathbf{a} = [a_1, \cdots, a_m]^T$ are unknown, bounded and s-periodic parameters without knowing the upper bounds. $\boldsymbol{\zeta}^0 = [\zeta_1^0, \cdots, \zeta_m^0]^T$ is a known vector valued local Lipschitz and continuously differentiable function w.r.t. arguments $\mathbf{x}$. $b(s) \in \mathcal{C}^1[0, \infty)$ is an unknown, bounded and s-periodic gain of the system input. The prior information about $b(s)$ is that $b(s)$ is positive for all s. Unknown parameters $a_i(s)$ may have different periods L_i for $i = 1, \cdots, m$ and $b(s)$ has a period L_{m+1}. In this section we assume that all the periods are rational numbers. In such circumstances, there exists a lowest common multiple L for all unknown coefficients $a_i(s)$ and $b(s)$. We can use the common period L as the updating period.

Remark 2.2 *In rotary machine systems, s represents the angular displacement. Due to the angular periodicity of 2π, most rotary machines will present certain cyclic behavior along the s axis. For example, due to the inherently spatially distributed structure in stators and rotors, most electrical motors produce torque ripples which can be modelled as spatially cyclic parameters in s domain. In our daily life and industry, we can observe many such examples as engines, turbines, gyros, conveyors, in which the rotary machine systems will inevitably produce periodic impacts that can be modelled as cyclic parameters or disturbances.*

Since the rotary system (2.11) is spatially periodic in s, we can first apply the ∇-operator (2.8), convert the dynamics from the t-domain to the s-domain, then design an appropriate controller with both feedback and periodic adaptation in the s-domain.

2.3.1 State transformation for high order systems by feedback linearization

Applying the ∇-operator to convert the system (2.11) from t domain to s domain, we have

$$\begin{cases} \nabla x_i &= \dfrac{x_{i+1}}{x_1}, \qquad i = 1, \cdots, n-1, \\ \nabla x_n &= \mathbf{a}^T(s)\boldsymbol{\zeta}(\mathbf{x}) + b(s)x_1^{-1}u. \end{cases} \tag{2.12}$$

where $\boldsymbol{\zeta} = \boldsymbol{\zeta}^0/x_1$. Note that (2.12) is not in canonical form. To facilitate the SPAC design, we can apply feedback linearization to transform the system (2.12) into a canonical form.

First define a state transformation $\mathbf{z} = \mathcal{T}(x_1, \cdots, x_n)$ as

$$z_1 = x_1, \qquad z_2 = \nabla x_1, \quad \cdots, \quad z_n = \nabla^{n-1} x_1, \tag{2.13}$$

where $\mathbf{z} = [z_1, \cdots, z_n]^T$, $\nabla^k = \nabla \cdot \nabla^{k-1}$. For a scalar function h with the arguments $x_1, \cdots, x_n$, denote the Lie derivative of h with respect to a vector $\mathbf{f} = [f_1, \cdots, f_n]^T$ as

$$L_{\mathbf{f}}h = L_{[f_1,\cdots,f_n]}h = \begin{bmatrix} \dfrac{\partial h}{\partial x_1}, & \cdots, & \dfrac{\partial h}{\partial x_n} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}. \tag{2.14}$$

The property of $\mathcal{T}$ is summarized in *Proposition* 2.3.

Proposition 2.3 *The transformation (2.13), which is a diffeomorphism, transforms the system (2.12) to the canonical form*

$$\begin{cases} \nabla z_i &= z_{i+1}, \qquad i = 1, \cdots, n-1, \\ \nabla z_n &= \mathbf{a}^T(s)\boldsymbol{\xi}^0(\mathbf{z}) + \rho(\mathbf{z}) + b(s)\eta(\mathbf{z})u, \end{cases} \tag{2.15}$$

where η, $\boldsymbol{\xi}^0$, ρ *are generated by substituting* $\mathbf{x} = \mathcal{T}^{-1}(\mathbf{z})$ *into functions* $\eta_x(\mathbf{x})$, $\boldsymbol{\xi}^0_x(\mathbf{x})$ *and* $\rho_x(\mathbf{x})$ *defined below*

$$\eta_x = \frac{1}{x_1^n}, \quad \boldsymbol{\xi}^0_x(\mathbf{x}) = x_1\eta_x\boldsymbol{\zeta}(\mathbf{x}), \quad \rho_x = \frac{x_1 L_{[x_2,\cdots,x_n]}N_{n-1} - (2n-3)x_2N_{n-1}}{x_1^{2n-1}},$$

N_{n-1} *is a polynomial with arguments* $x_1, \cdots, x_n$ *and its recursive form is given in the proof of this proposition.*

Proof: See *Appendix A.1.*

In speed regulation problems, x_1 is to track a given speed $x_{1,r}(t)$ which is generated by a reference model

$$\begin{cases} \dfrac{dx_{i,r}}{dt} &= x_{i+1,r}, \qquad i = 1, 2, \cdots, n-1, \\ \dfrac{dx_{n,r}}{dt} &= w^0(\mathbf{x}_r, r), \end{cases} \tag{2.16}$$

where $\mathbf{x}_r = [x_{1,r}, \cdots, x_{n,r}]^T$ is a vector of states, r is a constant reference input. To facilitate the controller design in the s domain, ∇-operator is applied to the reference model

$$\begin{cases} \nabla x_{i,r} &= \dfrac{x_{i+1,r}}{x_1}, \qquad i = 1, \cdots, n-1, \\ \nabla x_{n,r} &= \dfrac{w^0(\mathbf{x}_r, r)}{x_1}. \end{cases} \tag{2.17}$$

Define $\mathbf{z}_r = [z_{1,r}, \cdots, z_{n,r}]^T$ and the new state transformation $\mathbf{z}_r = \mathcal{T}_r(x_{1,r}, \cdots, x_{n,r})$

$$z_{1,r} = x_{1,r}, \qquad z_{2,r} = \nabla x_{1,r}, \qquad \cdots, \qquad z_{n,r} = \nabla^{n-1}x_{1,r}. \tag{2.18}$$

Analogous to the derivation procedure shown in *Proposition* 2.3, the state transformation (2.18) is a diffeomorphism and the inverse $\mathbf{x}_r = \mathcal{T}_r^{-1}(\mathbf{z}_r, \mathbf{z})$ exists. The reference model (2.17) is transformed into a new canonical model

$$\begin{cases} \nabla z_{i,r} &= z_{i+1,r}, \qquad i = 1, \cdots, n-1, \\ \nabla z_{n,r} &= w(\mathbf{z}_r, \mathbf{z}, r), \end{cases} \tag{2.19}$$

where $w(\mathbf{z}_r, \mathbf{z}, r)$ is generated by substituting $\mathbf{x} = \mathcal{T}^{-1}(\mathbf{z})$ and $\mathbf{x}_r = \mathcal{T}_r^{-1}(\mathbf{z}, \mathbf{z}_r)$ into the function $w_x(\mathbf{x}_r, \mathbf{x}, r)$ defined below

$$w_x(\mathbf{x}_r, \mathbf{x}, r) = \frac{x_1 L_{[x_2,\cdots,x_n,x_{3,r},\cdots,x_{n,r}]} N_{n-1,r} - (2n-3)x_2 N_{n-1,r}}{x_1^{2n-1}} + \frac{1}{x_1^{2n-2}} \frac{\partial N_{n-1,r}}{\partial x_{n,r}} w^0(\mathbf{x}_r, r).$$

and $N_{n-1,r}$ is a polynomial of arguments $\mathbf{x}$, $\mathbf{x}_r$ derived recursively in a similar way as N_{n-1} in *Proposition* 2.3.

To illustrative the spatial transformation, consider a 3rd order process. The state transformation $\mathcal{T}$ is $z_1 = x_1,\ z_2 = x_2/x_1,\ z_3 = (x_1x_3 - x_2^2)/x_1^3$, and the inverse $\mathcal{T}^{-1}$ is $x_1 = z_1,\ x_2 = z_1z_2,\ x_3 = z_1(z_1z_3 + z_2^2)$, or in matrix form

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & z_1 & 0 \\ 0 & z_1z_2 & z_1^2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}.$$

Similarly, the state transform and the inverse transform of the reference model, $\mathcal{T}_r$ and $\mathcal{T}_r^{-1}$ are respectively $z_{1,r} = x_{1,r},\ z_{2,r} = x_{2,r}/x_1,\ z_{3,r} = (x_1x_{3,r} - x_{2,r}x_2)/x_1^3$ and $x_{1,r} = z_{1,r},\ x_{2,r} = z_1z_{2,r},\ x_{3,r} = z_1^2z_{3,r} + z_1z_2z_{2,r}$.

2.3.2 Periodic adaptation and convergence analysis

Define the tracking error to be $\mathbf{e} = [e_1, \cdots, e_n]^T = \mathbf{z} - \mathbf{z}_r$. From the system (2.15) and the reference model (2.19), the error dynamics is

$$\begin{cases} \nabla e_i &= e_{i+1}, \qquad i = 1, \cdots, n-1, \\ \nabla e_n &= \mathbf{a}^T(s)\boldsymbol{\xi}^0(\mathbf{z}) + \rho(\mathbf{z}) + b(s)\eta(\mathbf{z})u - w(\mathbf{x}_r, \mathbf{z}, r), \end{cases} \tag{2.20}$$

or simply

$$\nabla \mathbf{e} = A\mathbf{e} + \mathbf{b}[\mathbf{a}^T\boldsymbol{\xi}^0 + (\sigma + \rho - w) + b\eta u], \tag{2.21}$$

where $\mathbf{b} = [0, \cdots, 0, 1]^T$, $\sigma = \mathbf{ce}$ and $\mathbf{c} = [c_1, \cdots, c_{n-1}, 1]$ is chosen such that

$$A \triangleq \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -c_1 & -c_2 & -c_3 & \cdots & -c_{n-1} & -1 \end{pmatrix} \tag{2.22}$$

is asymptotically stable. The periodic adaptive control mechanism is constructed below

$$u(s) = -\frac{1}{\eta(\mathbf{z})}\left[k\sigma(s) + \hat{\boldsymbol{\theta}}^T(s)\boldsymbol{\xi}(s,\mathbf{e})\right], \tag{2.23}$$

where $k > 0$ is a constant feedback gain, $\hat{\boldsymbol{\theta}} = [\hat{\theta}_1, \cdots, \hat{\theta}_{m+2}]^T$ is the estimate of the extended parametric vector

$$\boldsymbol{\theta}(s) = \left[\frac{1}{b}\mathbf{a}^T,\ \frac{1}{b},\ \frac{1}{b^2}\nabla b\right]^T$$

and

$$\boldsymbol{\xi}(s,\mathbf{e}) = \left[(\boldsymbol{\xi}^0)^T,\ \mathbf{c}_1\mathbf{e} + \rho - w,\ -\frac{\sigma}{2}\right]^T$$

where $\mathbf{c}_1 = [0, c_1, \cdots, c_{n-1}]$. Note that $\boldsymbol{\theta} \in \mathcal{C}^1([0,\infty); \mathcal{R}^{m+2})$.

The parametric updating law is

$$\begin{aligned} \hat{\boldsymbol{\theta}}(s) &= \hat{\boldsymbol{\theta}}(s-L) + \Gamma(s,L)\boldsymbol{\xi}(s)\sigma(s), \\ \hat{\boldsymbol{\theta}}(s) &= 0, \qquad \forall s \in [-L, 0], \end{aligned} \tag{2.24}$$

where $\Gamma > 0$ is the learning gain matrix defined in (2.1). The SPAC convergence property is summarized in Theorem 2.1.

Theorem 2.1 *For the system (2.15) and the reference model (2.19), the control law (2.23) and the periodic adaptation law (2.24) achieve the L-convergence of the tracking error* $\mathbf{e}$.

Proof. Substituting the learning control law (2.23) into (2.21) yields the closed-loop error dynamics

$$\begin{aligned} \nabla\mathbf{e} &= A\mathbf{e} + \mathbf{b}\left[\mathbf{a}^T\boldsymbol{\xi}^0 + (\sigma + \rho - w) + b\eta\left(-\frac{1}{\eta}(k\sigma + \hat{\boldsymbol{\theta}}^T\boldsymbol{\xi})\right)\right] \\ &= A\mathbf{e} + \mathbf{b}\left[-bk\sigma + \mathbf{a}^T\boldsymbol{\xi}^0 + (\sigma + \rho - w) - b\hat{\boldsymbol{\theta}}^T\boldsymbol{\xi}\right]. \end{aligned} \tag{2.25}$$

The error dynamics (2.25) and the parametric updating law (2.24) form a set of differential and continuous-space difference equations of neutral type. The existence of solution

for this class of systems has been discussed in [149]. Thus we focus on the convergence property.

Notice the facts $\mathbf{cb} = 1$, $\mathbf{c}A + \mathbf{c} = \mathbf{c}_1$ and $\sigma = \mathbf{ce}$, multiplying $\mathbf{c}$ on both sides of (2.25) yields

$$\begin{aligned} \mathbf{c}\nabla\mathbf{e} &= \mathbf{c}A\mathbf{e} + \mathbf{cb}\left[-bk\sigma + \mathbf{a}^T\boldsymbol{\xi}^0 + (\sigma + \rho - w) - b\hat{\boldsymbol{\theta}}^T\boldsymbol{\xi}\right] \\ &= -bk\sigma + \mathbf{a}^T\boldsymbol{\xi}^0 + (\mathbf{c}_1\mathbf{e} + \rho - w) - b\hat{\boldsymbol{\theta}}^T\boldsymbol{\xi}. \end{aligned} \tag{2.26}$$

First prove the convergence for $s \geq L$ by using the LKF

$$V(s) = \frac{1}{2b}\sigma^2 + \frac{1}{2}\int_{s-L}^{s} \boldsymbol{\phi}^T(\tau)\mathcal{B}^{-1}\boldsymbol{\phi}(\tau)d\tau, \tag{2.27}$$

where $\boldsymbol{\phi}(\tau) = \boldsymbol{\theta}(\tau) - \hat{\boldsymbol{\theta}}(\tau)$. The upper right hand derivative of V w.r.t. s is

$$\nabla V = \frac{1}{b}\sigma\mathbf{c}\nabla\mathbf{e} - \frac{1}{2b^2}\nabla b\cdot\sigma^2 + \frac{1}{2}\left[\boldsymbol{\phi}^T\mathcal{B}^{-1}\boldsymbol{\phi} - \boldsymbol{\phi}^T(s-L)\mathcal{B}^{-1}\boldsymbol{\phi}(s-L)\right] \tag{2.28}$$

where $\mathbf{c}\nabla\mathbf{e} = \nabla\sigma$. Substituting the dynamics (2.26) into the first two terms on the right hand side of (2.28)

$$\begin{aligned} \frac{1}{b}\sigma\mathbf{c}\nabla\mathbf{e} - \frac{1}{2b^2}\nabla b\cdot\sigma^2 &= \frac{1}{b}\sigma\left[-bk\sigma + \mathbf{a}^T\boldsymbol{\xi}^0 + (\mathbf{c}_1\mathbf{e} + \rho - w) - b\hat{\boldsymbol{\theta}}^T\boldsymbol{\xi}\right] - \frac{1}{2b^2}\nabla b\sigma^2 \\ &= -k\sigma^2 + \sigma\left[\frac{1}{b}\mathbf{a}^T\boldsymbol{\xi}^0 + \frac{1}{b}(\mathbf{c}_1\mathbf{e} + \rho - \omega) + \frac{1}{b^2}\nabla b(-\frac{\sigma}{2}) - \hat{\boldsymbol{\theta}}^T\boldsymbol{\xi}\right] \\ &= -k\sigma^2 + \sigma\left[\boldsymbol{\theta}^T\boldsymbol{\xi} - \hat{\boldsymbol{\theta}}^T\boldsymbol{\xi}\right] = -k\sigma^2 + \sigma\boldsymbol{\phi}^T\boldsymbol{\xi}. \end{aligned} \tag{2.29}$$

Applying the parametric adaptation law (2.24) where $\Gamma = \mathcal{B}$ for $s \geq L$, the periodic property $\boldsymbol{\theta}(s) = \boldsymbol{\theta}(s-L)$, and the algebraic relationship (2.10), the third term on the right-hand side of (2.28) is

$$\begin{aligned} &\frac{1}{2}\left[\boldsymbol{\phi}^T\mathcal{B}^{-1}\boldsymbol{\phi} - \boldsymbol{\phi}^T(s-L)\mathcal{B}^{-1}\boldsymbol{\phi}(s-L)\right] \\ &= \frac{1}{2}\left[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T\mathcal{B}^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) - \left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(s-L)\right)^T\mathcal{B}^{-1}\left(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}(s-L)\right)\right] \end{aligned}$$

$$= \frac{1}{2}\left[-2(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^T\mathcal{B}^{-1}\left(\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}}(s-L)\right)-\left(\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}}(s-L)\right)^T\mathcal{B}^{-1}\left(\hat{\boldsymbol{\theta}}-\hat{\boldsymbol{\theta}}(s-L)\right)\right]$$
$$= -\boldsymbol{\phi}^T\boldsymbol{\xi}\sigma-\frac{1}{2}\boldsymbol{\xi}^T\mathcal{B}\boldsymbol{\xi}\sigma^2. \tag{2.30}$$

It can be seen that the system uncertainty $\boldsymbol{\phi}^T\boldsymbol{\xi}\sigma$ appears on (2.29) and (2.30) with opposite signs. Thus by substituting (2.29) and (2.30) into (2.28), the upper right hand derivative of V is

$$\nabla V = -k\sigma^2-\frac{1}{2}\boldsymbol{\xi}^T\mathcal{B}\boldsymbol{\xi}\sigma^2 \le -k\sigma^2, \tag{2.31}$$

that is, ∇V is negative semi-definite for $s\in[L,\infty)$. From *Proposition* 2.1, we can derive the boundedness of σ and the L-convergence property $\lim_{s\to\infty}\int_{s-L}^{s}\sigma^2(\tau)d\tau = 0$ when $V(L)$ is finite. Notice the relationship

$$\sigma = \mathbf{ce} = e_n + c_{n-1}e_{n-1}+\cdots+c_1e_1 = (\nabla^{n-1}+c_{n-1}\nabla^{n-2}+\cdots+c_2\nabla+c_1)e_1,$$

where $\nabla^{n-1}+c_{n-1}\nabla^{n-2}+\cdots+c_2\nabla+c_1$ is a stable polynomial of the differential operator ∇. Therefore, the boundedness of σ implies the boundedness of $\mathbf{e}$, and the L-convergence of σ implies the L-convergence of $\mathbf{e}$.

Next prove the finiteness of $V(L)$. According to *Proposition* 2.2, we need only to prove the boundedness of $V(s)$ during the interval $[L_1, L)$, where $0 < L_1 \le L$. From (2.24) and the definition of the gain matrix (2.1), the adaptation law is

$$\hat{\boldsymbol{\theta}}(s) = \mathcal{A}(s)\boldsymbol{\xi}(s)\sigma(s) \tag{2.32}$$

or $\boldsymbol{\xi}\sigma = \mathcal{A}^{-1}\hat{\boldsymbol{\theta}}$. Define $V(s) = \frac{1}{2b}\sigma^2+\frac{1}{2}\int_0^s\boldsymbol{\phi}^T(\tau)\mathcal{B}^{-1}\boldsymbol{\phi}(\tau)d\tau$. Using the relationship (2.29), the upper right hand derivative of V is

$$\nabla V = -k\sigma^2+\boldsymbol{\phi}^T\boldsymbol{\xi}\sigma+\frac{1}{2}\boldsymbol{\phi}^T\mathcal{B}^{-1}\boldsymbol{\phi} = -k\sigma^2+\boldsymbol{\phi}^T\mathcal{A}^{-1}\hat{\boldsymbol{\theta}}+\frac{1}{2}\boldsymbol{\phi}^T\mathcal{B}^{-1}\boldsymbol{\phi} \tag{2.33}$$

By virtue of the analogy between (2.4) and (2.33), the boundedness of $V(L)$ is immediately obvious from *Proposition* 2.2. ■

2.4 SPAC for Systems with Pseudo-Periodic Parameters

In this section the parallel parametric adaptation is explored. The unknown parameter b is assumed to be an unknown positive constant.

From (2.21), the dynamics of the tracking error $\mathbf{e}$ is

$$\nabla \mathbf{e} = A\mathbf{e} + \mathbf{b}[\mathbf{a}^T\boldsymbol{\xi}^0 + (\sigma + \rho - w) + b\eta u] = A\mathbf{e} + b\mathbf{b}\left[\boldsymbol{\theta}^T\boldsymbol{\xi} + \mu(\sigma + \rho - w) + \eta u\right], \quad (2.34)$$

where $\boldsymbol{\theta} = b^{-1}\mathbf{a}$, $\boldsymbol{\xi} = \boldsymbol{\xi}^0$, $\mu = b^{-1}$. Using Lyapunov stability theory for LTI systems, for a given positive definite matrix $Q \in \mathcal{R}^{n\times n}$, there exists a unique positive definite matrix $P \in \mathcal{R}^{n\times n}$ satisfying the Lyapunov equation

$$A^T P + PA = -Q.$$

Denote λ_Q the minimum eigenvalue of the matrix Q such that $-\mathbf{x}^T Q\mathbf{x} \le -\lambda_Q \mathbf{x}^T\mathbf{x}$ for any $\mathbf{x} \in \mathcal{R}^n$. The spatial control mechanism is constructed as

$$u(s) = -\frac{1}{\eta}[\hat{\boldsymbol{\theta}}(s)^T\boldsymbol{\xi}^0 + \hat{\mu}(\sigma + \rho - w)], \quad (2.35)$$

where $\hat{\boldsymbol{\theta}}(s) = [\hat{\theta}_1, \cdots, \hat{\theta}_m]^T$ is the parameter estimate of $\boldsymbol{\theta}$ and $\hat{\mu}$ is the parameter estimate of μ. Note that we have periodic parameters $\boldsymbol{\theta}$ and time invariant parameter μ, hence use mixed periodic adaption and differential adaption laws

$$\begin{array}{rcll} \hat{\theta}_i(s) & = & \hat{\theta}_i(s - L_i) + \gamma_i(s, L_i)\xi_i(s)v(s), & \nabla\hat{\mu}(s) = \gamma[\sigma + \rho - w]v(s), \\ \hat{\theta}_i(s) & = & 0, \qquad \forall s \in [-L_i, 0], & i = 1, \cdots, m \end{array} \quad (2.36)$$

where $v(s) = \mathbf{b}^T P\mathbf{e}(s)$, L_i denotes the period of the unknown parameter a_i or θ_i, the adaptation gain $\gamma_i(s, L_i)$ is defined in (2.1), $\gamma > 0$ is a constant gain, ξ_i is the i-th entry of vector $\boldsymbol{\xi}$.

The SPAC convergence property is summarized in *Theorem* 2.2 below.

Theorem 2.2 *For the system (2.34), the spatial control mechanism (2.35) and (2.36) ensures the L-convergence of the tracking error* $\mathbf{e}$.

Proof. Substituting the spatial control law (2.35) into the dynamics (2.34) yields the closed-loop error dynamics

$$\begin{aligned}\nabla \mathbf{e} &= A\mathbf{e} + b\mathbf{b}[\boldsymbol{\theta}^T\boldsymbol{\xi} + \mu(\sigma+\rho-w) - \hat{\boldsymbol{\theta}}^T\boldsymbol{\xi} - \hat{\mu}(\sigma+\rho-w)] \\ &= A\mathbf{e} + b\mathbf{b}[\boldsymbol{\phi}^T\boldsymbol{\xi} + \psi(\sigma+\rho-w)], \end{aligned} \tag{2.37}$$

where $\boldsymbol{\phi} = [\phi_1, \cdots, \phi_m]^T = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$, $\psi = \mu - \hat{\mu}$. Define the LKF

$$V(s) = \frac{1}{2b}\mathbf{e}^T P\mathbf{e} + \frac{1}{2}\sum_{i=1}^{m}\frac{1}{\beta_i}\int_{\max\{0,s-L_i\}}^{s}\phi_i^2 d\tau + \frac{1}{2\gamma}\psi^2. \tag{2.38}$$

First consider the interval $[L,\infty)$ where $L = \max\{L_1, \cdots, L_m\}$. The upper right hand derivative of V w.r.t. s is

$$\nabla V = \frac{1}{2b}\left(\nabla\mathbf{e}^T P\mathbf{e} + \mathbf{e}^T P\nabla\mathbf{e}\right) + \frac{1}{2}\sum_{i=1}^{m}\frac{1}{\beta_i}\left[\phi_i^2(s) - \phi_i^2(s-L_i)\right] - \frac{1}{\gamma}\psi\nabla\hat{\mu}. \tag{2.39}$$

Substituting the dynamics (2.37) into the first term on the right-hand side of equation (2.39),

$$\begin{aligned}\frac{1}{2b}(\nabla\mathbf{e}^T P\mathbf{e} + \mathbf{e}^T P\nabla\mathbf{e}) &= \frac{1}{2b}\left[\mathbf{e}^T A^T + b\mathbf{b}^T\left(\boldsymbol{\phi}^T\boldsymbol{\xi} + \psi(\sigma+\rho-w)\right)\right]P\mathbf{e} \\ &\quad + \frac{1}{2b}\mathbf{e}^T P\left[A\mathbf{e} + b\mathbf{b}\left(\boldsymbol{\phi}^T\boldsymbol{\xi} + \psi(\sigma+\rho-w)\right)\right] \\ &= \frac{1}{2b}\mathbf{e}^T(A^T P + PA)\mathbf{e} + \boldsymbol{\phi}^T\boldsymbol{\xi}\mathbf{b}^T P\mathbf{e} + \psi(\sigma+\rho-w)\mathbf{b}^T P\mathbf{e} \\ &\leq -\frac{\lambda_Q}{2b}\|\mathbf{e}\|^2 + \boldsymbol{\phi}^T\boldsymbol{\xi}v + \psi(\sigma+\rho-w)v. \end{aligned} \tag{2.40}$$

The second term on the right-hand side of equation (2.39), by substituting the parametric updating law (2.36), the spatial periodicity $\theta_i(s) = \theta_i(s-L_i)$, and the algebraic relationship (2.10), is

$$\begin{aligned}\frac{1}{2}\sum_{i=1}^{m}\frac{1}{\gamma_i}\left[\phi_i^2 - \phi_i^2(s-L_i)\right] &= \frac{1}{2}\sum_{i=1}^{m}\frac{1}{\gamma_i}\left[\left(\theta_i - \hat{\theta}_i\right)^2 - \left(\theta_i - \hat{\theta}_i(s-L_i)\right)^2\right] \\ &= \sum_{i=1}^{m}[-(\theta_i - \hat{\theta}_i)\xi_i v - \frac{1}{2}\gamma_i(\xi_i v)^2] \\ &= -\boldsymbol{\phi}^T\boldsymbol{\xi}v - \frac{1}{2}\sum_{i=1}^{m}\gamma_i(\xi_i v)^2. \end{aligned} \tag{2.41}$$

The third term on the right-hand side of equation (2.39), by substituting the parametric updating law (2.36), becomes

$$-\frac{1}{\gamma}\psi\nabla\hat{\mu} = -\psi(\sigma+\rho-w)v. \tag{2.42}$$

The parametric uncertainties in (2.41) and (2.42) appear in (2.40) with opposite signs. As a result, substituting (2.40), (2.41) and (2.42) into (2.39) yields $\nabla V \leq -\frac{\lambda_Q}{2b}\|\mathbf{e}\|^2 \leq 0$, that is, ∇V is negative semi-definite for $s \geq L$. From *Proposition* 2.1, we can derive the boundedness of $\mathbf{e}$ and the L-convergence property $\lim_{s\to\infty}\int_{s-L}^{s}\|\mathbf{e}\|^2(\tau)d\tau = 0$ when $V(L)$ is finite.

The remaining is to prove the boundedness of $V(s)$ for $s \in [0, L]$. Without the loss of generality, assume the periods satisfy the relationship

$$L_1 < L_2 < \cdots < L_m = L,$$

and the interval $[0, L]$ is divided into m different sub-intervals $[L_j, L_{j+1}]$. Suppose $s \in [L_j, L_{j+1}]$, the LKF (2.38) renders to

$$\begin{aligned} V(s) &= \frac{1}{2b}\mathbf{e}^T P\mathbf{e} + \frac{1}{2}\sum_{i=1}^{j-1}\frac{1}{\beta_i}\int_{s-L_i}^{s}[\theta_i(\tau)-\hat{\theta}_i(\tau)]^2 d\tau \\ &\quad + \frac{1}{2}\sum_{i=j}^{m}\frac{1}{\beta_i}\int_{0}^{s}[\theta_i(\tau)-\hat{\theta}_i(\tau)]^2 d\tau + \frac{1}{2\gamma}(\mu-\hat{\mu})^2. \end{aligned} \tag{2.43}$$

The upper right hand derivative of the functional V w.r.t. s is

$$\begin{aligned} \nabla V &= \frac{1}{2b}\left(\nabla\mathbf{e}^T P\mathbf{e} + \mathbf{e}^T P\nabla\mathbf{e}\right) + \frac{1}{2}\sum_{i=1}^{j-1}\frac{1}{\beta_i}\left[\phi_i^2(s) - \phi_i^2(s-L_i)\right] \\ &\quad + \frac{1}{2}\sum_{i=j}^{m}\frac{1}{\beta_i}\phi_i^2(s) - \psi\nabla\hat{\mu}. \end{aligned} \tag{2.44}$$

The differential adaptation law for the constant parameter μ is the same for the entire time horizon $[0, \infty)$. On the other hand, the periodic parameter adaption (2.36) can be divided into two groups

$$\hat{\theta}_i(s) = \hat{\theta}_i(s-L_i) + \beta_i\xi_i(s)v(s) \quad i = 1, \cdots, j-1 \tag{2.45}$$

and

$$\hat{\theta}_i(s) = \alpha_i(s)\xi_i(s)v(s) \quad i = j, \cdots, m. \tag{2.46}$$

Since the parameter estimates and the LKF associated with parameters μ and θ_i, $i = 1, \cdots, j-1$, are the same as the preceding circumstance $s \geq L$, they appear in ∇V with negative semi-definite results. Thus we need only focus on the parameters θ_i, $i = j, \cdots, m$ and the upper right hand derivative of the LKF (2.43) is

$$\nabla V \leq -\frac{\lambda_Q}{2b}\|\mathbf{e}\|^2 + \sum_{i=j}^{m}\phi_i\xi_i v + \frac{1}{2}\sum_{i=j}^{m}\frac{1}{\beta_i}\phi_i^2(s). \tag{2.47}$$

For the ith term in (2.47), by substituting the adaptation law (2.46) we have

$$\phi_i\xi_i v + \frac{1}{\beta_i}\phi_i^2 = \frac{1}{\alpha_i}\phi_i\hat{\theta}_i + \frac{1}{\beta_i}\phi_i^2,$$

which is analogous to (2.4) as the scalar case. Therefore by applying *Proposition* 2.2 we can directly conclude the boundedness of ∇V in (2.47), in the sequel the finiteness of $V(s)$ in $[0, L_{j+1})$. ■

2.5 Illustrative Examples

Consider a simplified permanent magnet synchronous motor with a single link and under speed control [140]. The rotary part of the motor dynamics is

$$\begin{cases} \dfrac{ds}{dt} = x_1, \\ \dfrac{dx_1}{dt} = \frac{1}{J}(T_m - T_l - Bx_1), \end{cases} \tag{2.48}$$

where s and x_1 are motor angular position and speed, $J = 0.03\ kgm^2$ is the rated inertia, $B = 0.2$ is the unknown damping factor, T_m is the motor torque in the form

$$T_m = (1 + 0.2\cos 6s + 0.1\cos 12s)(1 + 0.5\cos s + 0.3\cos 2s)u\ Nm$$

with u the current. The $1st$ and $2nd$ harmonics in T_m are generated by the current measurement off-set and scaling errors [140]. The $6th$ and $12th$ harmonics in T_m are due to the distortion of the stator flux linkage distribution and variable magnetic reluctance at the stator slots. All harmonics are unknown. $T_l = \sin s \ Nm$ is the unknown load torque. Rewrite the system (2.48) in the form as (2.11)

$$\begin{cases} \dfrac{ds}{dt} &= x_1, \\ \dfrac{dx_1}{dt} &= \mathbf{a}^T(s)\boldsymbol{\zeta}^0(s, x_1) + b(s)u. \end{cases}$$

where

$$\mathbf{a}(s) = \left[-\frac{\sin s}{0.03} \quad -\frac{0.2}{0.03}\right]^T, \qquad \boldsymbol{\zeta}^0(s, x_1) = [1 \quad x_1]^T,$$

$$\text{and} \qquad b(s) = \frac{(1 + 0.2\cos 6s + 0.1\cos 12s)(1 + 0.5\cos s + 0.3\cos 2s)}{0.03}.$$

It can be seen that $m = 2$ and the updating period is 2π. The initial states are $\{s(0), x_1(0)\} = \{0, 0.1\}$. The reference motor speed is $x_{1,r} = 25 \ rad/s$. Choose $k = 0.05$, $\beta_i = 1$, $\alpha_i(s) = \sin(s/2 - \pi/2)/2 + 1/2$ for $i = 1, 2, 3, 4$. Applying the SPAC, the speed tracking error is shown in Fig.2.1. SPAC achieves satisfactory speed response despite the presence of rather large periodic parametric uncertainties in the state and input.

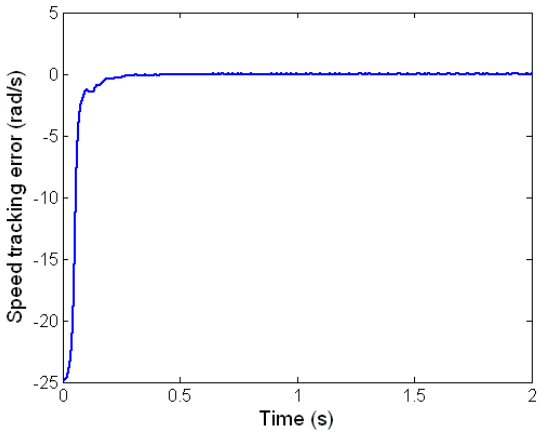


Figure 2.1: The speed tracking error profile in the time domain. The fast tracking convergence can be observed.

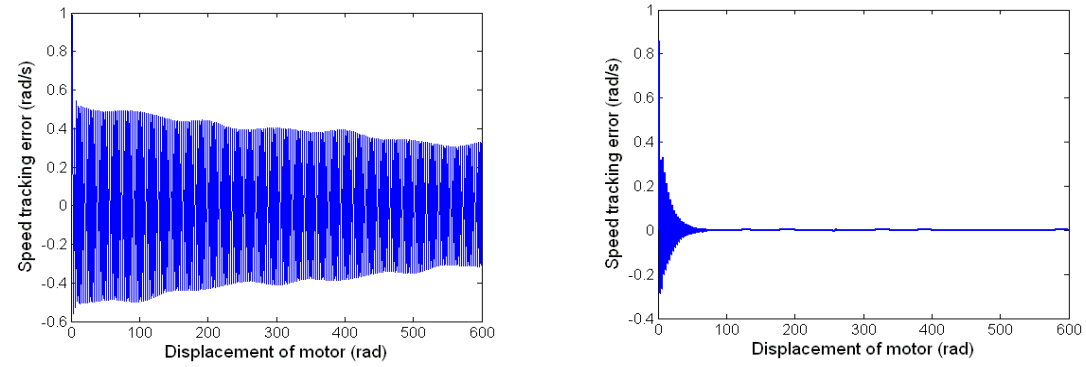
Next, to show the effect of SPAC with parallel adaptation for parameters with different periods, consider a dynamics

$$\begin{cases} \dfrac{ds}{dt} &= x_1, \\ \dfrac{dx_1}{dt} &= x_2, \\ \dfrac{dx_2}{dt} &= 2\sin\left(\frac{2\pi}{3}s\right)x_1^2 + 3\cos\left(\frac{2\pi}{100}s\right)x_2 + u. \end{cases} \tag{2.49}$$

The lowest common multiple is 300. The reference model is in the form of (2.16) with $n = 2$ and

$$\omega^0(\mathbf{x}_r, r) = -\frac{4\pi^2}{25}x_{1,r} + \frac{32\pi^2}{5},$$

which generates a sinusoidal reference speed $x_{1,r}$ in time domain with a period of 5 and magnitude of 3. Fig.2.2 (a) shows the error profile with $L = 300$, and (b) shows the error profile with two separate adaptation periods $L_1 = 3$ and $L_2 = 100$, both in the s domain. The performance improvement can be clearly seen by comparing the results in (a) and (b).



(a) SPAC with a common period $L = 300$.

(b) SPAC with two separate periods $L_1 = 3$ and $L_2 = 100$.

Figure 2.2: The speed tracking error profiles in the s domain. Parallel adaptation can effectively reduce the convergence time in SPAC.

When implementing the spatial adaptive control law with a digital controller, two practical issues arise: the sampling is done with a fixed time interval instead of a fixed spatial interval, and the memory size is finite.

Due to the time sampling and speed variation, quantities such as state variables at a specific position s may not be available, but available at two neighbouring positions s_1 and s_2, $s_1 < s < s_2$, corresponding to two adjacent sampling instances. Consequently $\hat{\theta}(s-L)$ may not be available when updating $\hat{\theta}(s)$. In such circumstances, $\hat{\theta}(s-L)$ can be interpolated by $\hat{\theta}(s_1-L)$ and $\hat{\theta}(s_2-L)$, or replaced by either of the neighbouring points when the sampling frequency is sufficiently higher than the process bandwidth.

Analogously, a digital controller can only store a finite number of data due to a finite buffer. Since the periodicity and sampling interval are fixed, it is adequate to choose the memory size equal or greater than the quotient equal to the periodicity divided by the sampling interval.

2.6 Conclusion

In this chapter a SPAC approach was proposed for rotary machines by using the repetitiveness of system in spatial domain. The SPAC can achieve L-convergence for any rotary machines systems that have spatially periodic parameter uncertainties or disturbances. The spatial periodic adaptation mechanism can work well even through the spatially periodic parameters may be aperiodic along the time axis. The main contributions of this work were to provide a feedback linearization method for high order rotary systems, and extend the periodic adaptation to pseudo-periodic parameters without a common period. The Lyapunov-Krasovskii functional provides a useful mathematical tool for SPAC property analysis.

In the next chapter, we will consider the AC with lifting technique for a more complex system structure in which the uncertainties show periodic repetitiveness in time domain.

Chapter 3

Discrete-Time Adaptive Control for Nonlinear Systems with Periodic Parameters: A Lifting Approach

3.1 Introduction

Periodic variations are invariant under a shift by one or more periods. They are often a consequence of some rotational motion at constant speed, and encountered in many real systems such as electrical motors, generators, helicopter blades and satellites [27, 28, 31, 61, 64, 95, 127, 150]. As in the case of linear periodic systems, many results have been achieved to deal with their adaptive control, robustness and identification [54, 91, 120]. Recently, discrete-time periodic adaptive control (PAC) has been proposed and the underlying idea of PAC is to update parameters in the same instance of two consecutive periods [1, 45]. Due to the time-varying nature, it would be very difficult, if not impossible, to design appropriate periodic adaptive controllers for more general scenarios such as plants with unknown control directions, plants in parametric-strict-

feedback form, plants not satisfying any growth conditions, etc.

On the other hand, many effective adaptive control methods have been developed for discrete-time systems with time-invariant parametric uncertainties, such as [166] for parametric-strict-feedback form, [39, 74] for unknown control direction, [65] for plants without any growth conditions in nonlinearities. It would be highly desirable if we can apply these well established adaptive control methods to plants with periodic parameters. To achieve this objective, we adopt a lifting technique to convert periodic parameters into an augmented vector of time-invariant parameters, in the sequel all existing adaptive control methods can be applied.

The underlying idea of the proposed approach is shown in Fig. 3.1. Denote θ_k^o a periodic unknown parameter with a periodicity $\mathcal{N}$ that is $\theta_k^o = \theta_{k-\mathcal{N}}^o$ where $\mathcal{N}$ is a positive integer, and ξ_k a known nonlinear regressor of the system states. Figure 3.1 shows that the product $\theta_k^o \xi_k$ can be converted to the product of two augmented vectors $\boldsymbol{\theta}^T \boldsymbol{\xi}_k$ where $\boldsymbol{\theta}$ consists of constant unknowns, and $\boldsymbol{\xi}_k$ has only one non-trivial element at each time k. From the figure, we can see that the idea is to extend the periodic θ_k^o to N constant unknown parameters, and meanwhile let the position of ξ_k rotate in the augmented vector $\boldsymbol{\xi}_k$, such that the equality $\theta_k^o \xi_k = \boldsymbol{\theta}^T \boldsymbol{\xi}_k$ holds for every time instant. In this way, the time-varying parametric uncertainties are simplified into time-invariant ones, while the known nonlinear regressor is changed to a more sophisticated structure. Note that the increasing complexity in the augmented regressor vector $\boldsymbol{\xi}_k$ with the structural rotation does not hinder the adaptive controller designs because the augmented vector is known. As a consequence adaptive control methods developed for time-invariant parameters can be applied to periodic cases by reformulating the problem with the lifting and conversion.

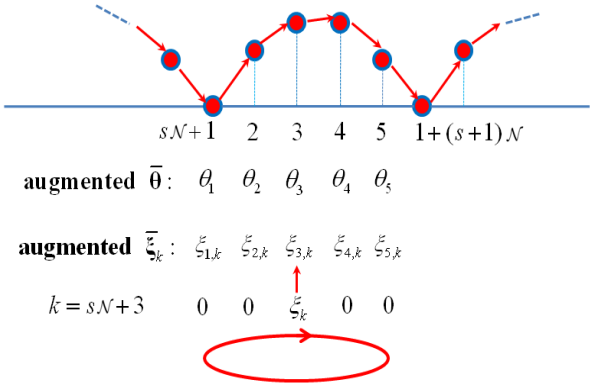


Figure 3.1: The concept of the proposed approach that converts periodic parameters into time-invariant ones using the lifting technique. Let the original periodic parameter be θ_k^o with a periodicity $\mathcal{N} = 5$, then θ_k^o has at most five distinguished constant values, denoted by a augmented vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]^T$. Let the known regressor be ξ_k, it can be extended to an augmented vector-valued regressor $\boldsymbol{\xi}_k = [\xi_{1,k}, \xi_{2,k}, \xi_{3,k}, \xi_{4,k}, \xi_{5,k}]^T$, in which there is only one non-trivial element and the remaining four are zeros at every time instance k. The non-trivial element locates at 3rd position when $k = s\mathcal{N} + 3$, $s = 0, 1, \cdots$, and in general at jth position when $k = s\mathcal{N} + j$. As the time k evolves, the position of the non-trivial element will keep rotating rightwards, returning from the rightmost position to the leftmost position, and starting over again. It is easy to verify the equality $\theta_k^o \xi_k = \boldsymbol{\theta}^T \boldsymbol{\xi}_k$.

The lifting technique has been used to convert periodic parametric uncertainties into augmented time-invariant ones [83, 84]. The results developed hitherto are limited to linear systems. In this work, the lifting technique is applied to nonlinear plants and in particular to adaptive control problems.

It is worthwhile pointing out some existing adaptive control approach developed for time-varying parameters [169, 172–174]. The main differences between their results and the results in this work lie in that theirs do not require the periodic condition and achieve a bounded tracking error that would vanish when the parameters become time-invariant.

On the contrary, in our work we require the parameters be periodic, but achieve the asymptotic convergence property.

It is also worthwhile highlighting the differences between the PAC approach and the approach proposed in this work. Due to the difficulty in dealing with time-varying parameters by adaptive control, in PAC [1] the updating period is chosen to be the least common multiple for all periodic parameters. This way simplifies the control problem and facilitates the convergence analysis for PAC, but also leads to slower convergence as the common period could be much larger than individual periods. For instance, the common period could be as large as 1155 while the periods of four parameters are respectively 3, 5, 7 and 11. By using the proposed adaptive control approach with lifting, parametric updating can be carried out in parallel for the four parameters, which is more than 100 times shorter than the common period.

The Chapter is organized as follows. In Section 3.2, we present the lifting technique which converts time-varying parameters into time-invariant ones and verify the equivalence between the proposed adaptive control law and the PAC law. Sections 3.3 extends the lifting approach to general nonlinear plants with multiple unknown parameters, periodic input gain, nonlinear growth condition, nonlinear parameterization, and tracking tasks. Section 3.4 extends the lifting approach to plants with parametric-strict-feedback structure and periodic unknown input gains. To the end, several illustrative examples are provided in Section 3.5.

3.2 Problem Formulation and Lifting Approach

In order to clearly illustrate the underlying idea of the new lifting approach, we focus on an one-dimensional system in this section and consider the regulation problem

$$x_{k+1} = \theta_k^o \xi_k + u_k, \tag{3.1}$$

where $\theta_k^o \in \mathcal{R}^1$ is periodic with the periodicity $\mathcal{N} \geq 1$, namely $\theta_k^o = \theta_{k-\mathcal{N}}^o$, and the nonlinear regressor $\xi_k = \xi(x_k)$ satisfies the Lipschitz continuity or sector-bounded condition. Note that, when $\mathcal{N} = 1$, the periodic parameter renders to time-invariant because $\theta_k^o = \theta_{k-1}^o$. Therefore PAC is the generalization of adaptive control that deals with time-invariant parameters.

3.2.1 Discrete-time PAC revisited

PAC designs for $\mathcal{N} \geq 2$ are presented in [1] and [45], and briefly summarized below

$$u_k = -\hat{\theta}_k \xi_k, \quad \hat{\theta}_k = \hat{\theta}_{k-\mathcal{N}} + \frac{x_{k-\mathcal{N}+1}}{c + \xi_{k-\mathcal{N}}^2} \xi_{k-\mathcal{N}}, \quad \hat{\theta}_j = \epsilon_0, \quad j \leq \mathcal{N} - 1, \tag{3.2}$$

where $c > 0$ and ϵ_0 is an arbitrary constant. The stability of PAC (3.2) for system (3.1) can be derived as follows. Substituting (3.2) into (3.1), we have

$$x_{k+1} = \tilde{\theta}_k \xi_k. \tag{3.3}$$

Since $\theta_k^o = \theta_{k-\mathcal{N}}^o$, using (3.2) and (3.3) yields

$$\tilde{\theta}_k \;=\; \tilde{\theta}_{k-\mathcal{N}} - \frac{x_{k-\mathcal{N}+1}\xi_{k-\mathcal{N}}}{c + \xi_{k-\mathcal{N}}^2} = \tilde{\theta}_{k-\mathcal{N}} - \frac{\xi_{k-\mathcal{N}}^2}{c + \xi_{k-\mathcal{N}}^2} \tilde{\theta}_{k-\mathcal{N}} = \frac{c}{c + \xi_{k-\mathcal{N}}^2} \tilde{\theta}_{k-\mathcal{N}}.$$

Next consider the incremental change of $\tilde{\theta}_k$ between two cycles

$$\begin{aligned} \tilde{\theta}_{k+\mathcal{N}}^2 - \tilde{\theta}_k^2 &= \left(\frac{c^2}{(c + \xi_k^2)^2} - 1 \right) \tilde{\theta}_k^2 \\ &= \left(\frac{c^2}{(c + \xi_k^2)^2} - 1 \right) \left(\frac{c + \xi_k^2}{\xi_k^2} \right) \frac{\xi_k^2 \tilde{\theta}_k^2}{c + \xi_k^2} \\ &= -\frac{2c + \xi_k^2}{c + \xi_k^2} \frac{\xi_k^2 \tilde{\theta}_k^2}{c + \xi_k^2}. \end{aligned} \tag{3.4}$$

We observe that $\tilde{\theta}_k^2$ is a bounded nonincreasing function. Summarizing (3.4), we have

$$\left\|\tilde{\theta}_k\right\|^2 = \left\|\tilde{\theta}_{k-j\mathcal{N}}\right\|^2 - \sum_{s=1}^{j} \frac{2c+\xi_{k-s\mathcal{N}}^2}{c+\xi_{k-s\mathcal{N}}^2} \frac{x_{k-s\mathcal{N}+1}^2}{c+\xi_{k-s\mathcal{N}}^2}$$

where $k - j\mathcal{N} \in [0, \mathcal{N})$ and $j = [\frac{k}{\mathcal{N}}]$ is an integer equal to or nearest to $k/\mathcal{N}$ from below. Since $\tilde{\theta}_k^2$ is nonnegative, we can conclude $\lim_{k\to\infty} \sum_{s=1}^{[\frac{k}{\mathcal{N}}]} \frac{x_{k-s\mathcal{N}+1}^2}{c+\xi_{k-s\mathcal{N}}^2} < \infty$. Therefore, $\lim_{k\to\infty} \frac{x_{k+1}^2}{c+\xi_k^2} = 0$, and by the Key Technique Lemma $x_k \to 0$ as $k \to \infty$.

3.2.2 Proposed lifting approach

In this part we present the lifting technique to convert periodic parameters to time-invariant ones. This idea can be applied to general discrete-time systems with any periodic parametric uncertainties. Subsequently, we can simply construct well established adaptive controllers that are developed for time-invariant parameters, even though the original plant parameters are periodic in essence.

Consider a sequence of finite intervals with the length of $\mathcal{N} > 0$. Let $j \in \{1, \cdots, \mathcal{N}\}$, then any time instance can be denoted to be $k = s\mathcal{N} + j$, $s = 0, 1, 2, \cdots$. Define an augmented parametric vector $\boldsymbol{\theta} = [\theta_1, \ \cdots, \ \theta_{\mathcal{N}}]^T \in \mathcal{R}^{\mathcal{N}}$, and a vector-valued regressor function $\boldsymbol{\xi}_k = \left[\ 0, \ \cdots, \ 0, \ \xi_k, \ 0, \ \cdots, \ 0\ \right]^T \in \mathcal{R}^{\mathcal{N}}$ for $k = s\mathcal{N} + j$, where ξ_k is the jth element of the regressor $\boldsymbol{\xi}_k$ and the only element that may not be zero at the time instance k. It can be seen that the position of ξ_k rotates in the regressor $\boldsymbol{\xi}_k$. The jth element of $\boldsymbol{\xi}_k$ will be ξ_k for $k = s\mathcal{N} + j$, $s = 0, 1, 2, \cdots$, and zero for the rest of time. In other words, ξ_k rotates in $\boldsymbol{\xi}_k$ with the periodicity of $\mathcal{N}$.

Notice the fact $\theta_k^o = \theta_{s\mathcal{N}+j}^o = \theta_j$ by virtue of the periodicity $\mathcal{N}$, and $\boldsymbol{\xi}_k$ has only the j-th element, we have $\boldsymbol{\theta}^T\boldsymbol{\xi}_k = \theta_j\xi_k = \theta_k^o\xi_k$, and the system (3.1) can be rewritten as

$$x_{k+1} = \boldsymbol{\theta}^T\boldsymbol{\xi}_k + u_k. \tag{3.5}$$

The uncertainty in (3.5) is time invariant and the nonlinear regressor $\boldsymbol{\xi}_k$ satisfies the

Lipschitz continuity or sector-bounded condition only if the original ξ_k does. This implies that the original problem with periodic parameters has been converted to the time-invariant problem, hence various existing adaptive control methods for time-invariant parametric uncertainties can be applied directly. For the system (3.5), the classical adaptive control law is

$$u_k = -\hat{\boldsymbol{\theta}}_k^T \boldsymbol{\xi}_k, \quad \hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \frac{x_k \boldsymbol{\xi}_{k-1}}{c + \|\boldsymbol{\xi}_{k-1}\|^2}, \; c > 0, \; \hat{\boldsymbol{\theta}}_0 = \epsilon_0 [1, \cdots, 1]^T, \tag{3.6}$$

where $\hat{\boldsymbol{\theta}}_k = [\hat{\theta}_{1,k}, \cdots, \hat{\theta}_{j,k}, \cdots, \hat{\theta}_{\mathcal{N},k}]^T$.

Theorem 3.1 *The adaptive control law (3.6) with the lifting approach, and the PAC law (3.2), are equivalent.*

Proof: Consider time instances $k = s\mathcal{N} + j$, $s = 0, 1, \cdots$, $j = 1, \cdots, \mathcal{N}$. For a given j, the only non-zero elements in $\boldsymbol{\xi}_k$ is the jth element, ξ_k, hence $\hat{\boldsymbol{\theta}}_k^T \boldsymbol{\xi}_k = \hat{\theta}_{j,k} \xi_k$. If we can prove $\hat{\theta}_{j,k} = \hat{\theta}_k$, then $\hat{\boldsymbol{\theta}}_k^T \boldsymbol{\xi}_k = \hat{\theta}_{j,k} \xi_k = \hat{\theta}_k \xi_k$.

Using the fact $\|\boldsymbol{\xi}_k\|^2 = \xi_k^2$, from the updating law (3.6) we obtain

$$\hat{\boldsymbol{\theta}}_{k-\mathcal{N}+1} \;=\; \hat{\boldsymbol{\theta}}_{k-\mathcal{N}} + \frac{x_{k-\mathcal{N}+1}}{c + \xi_{k-\mathcal{N}}^2} \boldsymbol{\xi}_{k-\mathcal{N}}. \tag{3.7}$$

Since $k - \mathcal{N} = (s-1)\mathcal{N} + j$, the only non-zero element in the vector-valued regressor $\boldsymbol{\xi}_{k-\mathcal{N}}$ is the jth element, $\xi_{k-\mathcal{N}}$, while other elements are zero. Thus among $\mathcal{N}$ parameters in (3.7), the only one updated at $k - \mathcal{N} + 1$ is the jth component, namely

$$\hat{\theta}_{j,k-\mathcal{N}+1} \;=\; \hat{\theta}_{j,k-\mathcal{N}} + \frac{x_{k-\mathcal{N}+1}}{c + \xi_{k-\mathcal{N}}^2} \xi_{k-\mathcal{N}}. \tag{3.8}$$

Since the non-zero element in the vector-valued regressor rotates with the periodicity of $\mathcal{N}$, from $k - \mathcal{N} + 1$ to k the element at the jth place is zero. In other words, from $k - \mathcal{N} + 1$ to k, there is no updating for the jth parameter, thus

$$\hat{\theta}_{j,k-\mathcal{N}+1} = \hat{\theta}_{j,k-\mathcal{N}+2} = \cdots = \hat{\theta}_{j,k}. \tag{3.9}$$

Substituting the relationship (3.9) into (3.8) yields

$$\hat{\theta}_{j,k} = \hat{\theta}_{j,k-\mathcal{N}} + \frac{x_{k-\mathcal{N}+1}}{c + \xi_{k-\mathcal{N}}^2}\xi_{k-\mathcal{N}} \tag{3.10}$$

which is the same as (3.2). By choosing the same initial conditions for (3.2) and (3.10), we have $\hat{\theta}_{j,k} = \hat{\theta}_k$. ■

3.3 Extension to General Cases

In this section, we explore possible extensions to various scenarios with multiple periodic parameters, periodic input gain, nonlinear growth condition, nonlinear parameterization, tracking task, respectively. Meanwhile, the advantage of the lifting approach will be made clear.

3.3.1 Extension to multiple parameters and periodic input gain

For simplicity, consider a scalar system

$$x_{k+1} = (\boldsymbol{\theta}_k^o)^T \boldsymbol{\xi}_k + b_k^o u_k, \quad x(0) = x_0, \tag{3.11}$$

where $\boldsymbol{\theta}_k^o = [\theta_{1,k}^o, \cdots, \theta_{m,k}^o]^T$ are unknown periodic parameters, $\boldsymbol{\xi}_k = [\xi_{1,k}, \cdots, \xi_{m,k}]^T$ is a known vector-valued regressor, and $b_k^o \in C[0, \infty)$ is a periodic uncertain gain of the system input. Note that each unknown parameter, $\theta_{i,k}^o$ or b_k^o, may have its own period $\mathcal{N}_i$ or $\mathcal{N}_b$. To avoid control singularity, it is assumed that b_k^o has a lower bound, that is, $b_k^o \geq b_{min}$ where $b_{min} > 0$ is known. The PAC designs will still be applicable by using the least common multiple of $\mathcal{N}_i$ and $\mathcal{N}_b$ as the updating period $\mathcal{N}$ [1, 45]. However, the use of the common period will make the periodic adaptation inefficient. If possible, the periodic adaptation should be conducted according to individual periods. To address this issue, we reconsider system (3.11) by using the lifting approach. The presence of

the uncertain periodic input gain presents another difficulty for PAC designs and leads to complicated controllers as shown in [45]. By converting the periodic parameters into time-invariant, the adaptive control design can be simplified.

To derive the lifting adaptive control law, first define the augmented parametric vector and corresponding vector-valued nonlinearity regressor. Note that the i-th element of $\boldsymbol{\theta}_k^o$ is periodic with period $\mathcal{N}_i$, thus there exist $\mathcal{N}_i$ values $[\theta_{i,1}, \cdots, \theta_{i,\mathcal{N}_i}]^T$. We can construct an augmented vector including all m periodic parameters

$$\begin{aligned} \bar{\boldsymbol{\theta}} &\triangleq \left[\bar{\boldsymbol{\theta}}_1^T, \cdots, \bar{\boldsymbol{\theta}}_m^T\right]^T \\ &= [\theta_{1,1}, \cdots, \theta_{1,\mathcal{N}_1}, \theta_{2,1}, \cdots, \theta_{m,1}, \cdots, \theta_{m,\mathcal{N}_m}]^T \in \mathcal{R}^{\mathcal{N}_1+\cdots+\mathcal{N}_m}, \end{aligned} \tag{3.12}$$

with all elements being constant. Accordingly, we can define an augmented regressor

$$\bar{\boldsymbol{\xi}}_k = \left[\bar{\boldsymbol{\xi}}_{1,k}^T, \cdots, \bar{\boldsymbol{\xi}}_{m,k}^T\right]^T \in \mathcal{R}^{\mathcal{N}_1+\cdots+\mathcal{N}_m}, \tag{3.13}$$

where $\bar{\boldsymbol{\xi}}_{i,k} = [0, \cdots, 0, \xi_{i,k}, 0, \cdots, 0]^T \in \mathcal{R}^{\mathcal{N}_i}$, and the element $\xi_{i,k}$ appears in the jth position of $\bar{\boldsymbol{\xi}}_{i,k}$ only if $k = s\mathcal{N}_i + j$, for $j = 1, 2, \cdots, \mathcal{N}_i$. It can be seen that m functions $\xi_{i,k}$, $i = 1, \cdots, m$, rotate according to their own periodicity, $\mathcal{N}_i$, respectively. As a result, for each time instance k, we have

$$(\boldsymbol{\theta}_k^o)^T \boldsymbol{\xi}_k = \bar{\boldsymbol{\theta}}^T \bar{\boldsymbol{\xi}}_k \tag{3.14}$$

which converts periodic parameters into an augmented time-invariant vector.

Analogously we convert b_k^o into an augmented vector $\mathbf{b} = [b_1, \cdots, b_{\mathcal{N}_b}]^T$ and meanwhile define a regressor $\boldsymbol{\zeta}_k = [0, \cdots, 0, 1, 0, \cdots, 0]^T \in \mathcal{R}^{\mathcal{N}_b}$, where the element 1 appears in the jth position of $\boldsymbol{\zeta}_k$ only when $k = s\mathcal{N}_b + j$. Hence for every time instance $b_k^o = \mathbf{b}^T \boldsymbol{\zeta}_k$, i.e. b_k^o is converted into an augmented time-invariant parametric vector. Consequently the system (3.11) can be rewritten as below

$$x_{k+1} = \bar{\boldsymbol{\theta}}^T \bar{\boldsymbol{\xi}}_k + \mathbf{b}^T \boldsymbol{\zeta}_k u_k, \quad x(0) = x_0. \tag{3.15}$$

where all parametric uncertainties in (3.15) are time invariant.

Now we are in a position to apply existing adaptive control methods. First consider a specific case where b_k^o is a known constant. This problem has been addressed in [166]. Second consider another specific case where b_k^o is an unknown constant. This problem has been solved by introducing a discrete Nussbaum gain [74]. Third consider the periodic b_k^o and only the control direction is assumed known. Using the least common multiple of $\mathcal{N}_i$ and $\mathcal{N}_b$, the PAC [45] can handle this problem but the adaptation speed would be slow. By using the lifting approach, we can address parametric uncertainties in (3.15) in a more efficient way. Note that none of the existing discrete-time adaptive control methods consider the scenario where the input gain is the product of two vectors $\mathbf{b}^T\boldsymbol{\zeta}_k$ as in (3.15). Thus in the following we extend the preceding adaptive controller design (3.6) to the generic system (3.15).

Define $\hat{\boldsymbol{\phi}}_k = [\hat{\boldsymbol{\theta}}_k^T,\ \hat{\mathbf{b}}_k^T]^T$ be the estimation of time-invariant parametric uncertainties $\bar{\boldsymbol{\theta}}$ and $\mathbf{b}$, $\tilde{\boldsymbol{\phi}}_k = [\tilde{\boldsymbol{\theta}}_k^T,\ \tilde{\mathbf{b}}_k^T]^T$ be the estimation errors where $\tilde{\boldsymbol{\theta}}_k = \bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}_k,\ \tilde{\mathbf{b}}_k = \mathbf{b} - \hat{\mathbf{b}}_k$, and $\boldsymbol{\xi}_k^* = \left[\bar{\boldsymbol{\xi}}_k^T, -\frac{\hat{\boldsymbol{\theta}}_k^T\bar{\boldsymbol{\xi}}_k}{\hat{\mathbf{b}}_k^T\boldsymbol{\zeta}_k}\boldsymbol{\zeta}_k^T\right]^T$. Design the adaptive control law

$$u_k = -\frac{\hat{\boldsymbol{\theta}}_k^T\bar{\boldsymbol{\xi}}_k}{\hat{\mathbf{b}}_k^T\boldsymbol{\zeta}_k}, \tag{3.16}$$

$$\hat{\boldsymbol{\phi}}_k = L\left[\hat{\boldsymbol{\phi}}_{k-1} + \frac{x_k\boldsymbol{\xi}_{k-1}^*}{c + \|\boldsymbol{\xi}_{k-1}^*\|^2}\right],\ \ c > 0,\ \ \hat{\boldsymbol{\phi}}(0) = \epsilon_0 I, \tag{3.17}$$

where the projector L is to guarantee each element of $\hat{\mathbf{b}}_k$ not below the bound b_{min}. Denote $[\mathbf{a}_{1,k}^T, \mathbf{a}_{2,k}^T]^T$ the vector $\hat{\boldsymbol{\phi}}_k$, where $\mathbf{a}_{1,k}$ is the update of $\hat{\boldsymbol{\theta}}_k$, and $\mathbf{a}_{2,k}$ is the update of $\hat{\mathbf{b}}_k$. By the projector we have $L[\mathbf{a}_{1,k}] = \mathbf{a}_{1,k}$, and for each element a of $\mathbf{a}_{2,k}$

$$L[a] = \begin{cases} a & a > b_{min} \\ a_{min} & a \le b_{min}. \end{cases}$$

The convergence analysis can be conducted similarly as in the PAC [1] by choosing the periodicity $\mathcal{N} = 1$.

The performance improvement of the lifting approach based adaptive control can be seen from the construction of the vectors $\bar{\boldsymbol{\xi}}_k$, where the element $\xi_{i,k}$ appears in the jth position of $\bar{\boldsymbol{\xi}}_{i,k}$ only if $k = s\mathcal{N}_i+j$, for $j = 1, 2, \cdots, \mathcal{N}_i$. The m regression components, $\xi_{i,k}$, $i = 1, \cdots, m$, rotate according to their own periodicity, $\mathcal{N}_i$, respectively. In other words, the parameter estimate $\hat{\theta}_{i,k}$ will be updated repeatedly after every $\mathcal{N}_i$ steps, namely updated according to its own periodicity. The same is for the update of $\hat{b}_k$ because the only non-zero element, which is 1, rotates in the augmented regressor $\boldsymbol{\zeta}_k$ and returns to the same position after $\mathcal{N}_b$ steps. Figure 2 illustrates the concept of the lifting method. Here $m = 1$, $\mathcal{N}_1 = 3$, $\mathcal{N}_b = 2$, therefore 5 parameters are updated.

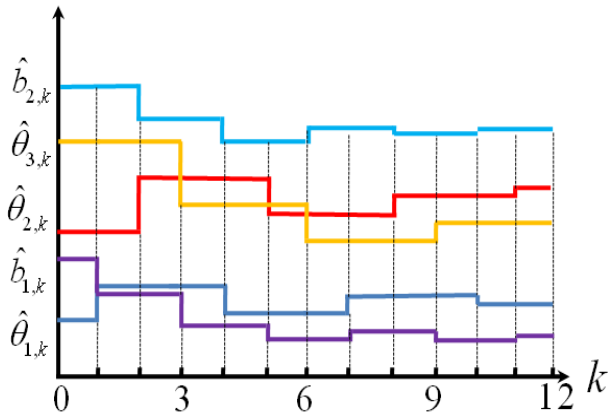


Figure 3.2: Illustration of lifting based concurrent adaptation law (3.17) with the periodicities $\mathcal{N}_1 = 3$ and $\mathcal{N}_b = 2$. It can be seen that $\hat{\theta}_{1,k}$ is updated at $k = s \times 3 + 1$, i.e. $k = 1, 4, \cdots$; $\hat{\theta}_{2,k}$ is updated at $k = s \times 3 + 2$, i.e. $k = 2, 5, \cdots$; $\hat{\theta}_{3,k}$ is updated at $k = s \times 3 + 3$, i.e. $k = 3, 6, \cdots$; $\hat{b}_{1,k}$ is updated at $k = s \times 2 + 1$, i.e. $k = 1, 3, \cdots$; and $\hat{b}_{2,k}$ is updated at $k = s \times 2 + 2$, i.e. $k = 2, 4, \cdots$.

Next consider in (3.11) a more generic scenario where the sign of the periodic b_k^o, i.e. the control direction, is unknown to us. For simplicity we consider $m = 1$, i.e. $\theta_k^o \xi_k$ that

can be expressed as $\boldsymbol{\theta}^T\boldsymbol{\xi}_k$, and

$$x_{k+1} = \boldsymbol{\theta}^T\boldsymbol{\xi}_k + b_k^o u_k = b_k^o\left[(b_k^o)^{-1}\boldsymbol{\theta}^T\boldsymbol{\xi}_k + u_k\right]. \tag{3.18}$$

Define a new augmented parametric vector $\bar{\boldsymbol{\theta}} = \left[b_1^{-1}\boldsymbol{\theta}^T, \cdots, b_{\mathcal{N}_b}^{-1}\boldsymbol{\theta}^T\right]^T \in \mathcal{R}^{\mathcal{N}_b\mathcal{N}}$, and a new regressor $\bar{\boldsymbol{\xi}}_k = \left[0, \cdots, 0, \boldsymbol{\xi}_k^T, 0, \cdots, 0\right]^T \in \mathcal{R}^{\mathcal{N}_b\mathcal{N}}$, where the only non-zero element, ξ_k, appears at the jth place for $k = s\mathcal{N}_b\mathcal{N} + j$, $j = 1, 2, \cdots, \mathcal{N}_b\mathcal{N}$, $s = 0, 1, \cdots$, and rotates with the periodicity of $\mathcal{N}_b\mathcal{N}$. The model (3.18) can be rewritten to be

$$x_{k+1} = b_k^o\left[\bar{\boldsymbol{\theta}}^T\bar{\boldsymbol{\xi}}_k + u_k\right]. \tag{3.19}$$

Since the sign of b_k^o is unknown, discrete Nussbaum gain scheme must be adopted. Similar to the work [162], a discrete-time adaptive control law with Nussbaum gain can be constructed to solve regulation problems

$$\begin{aligned} u_k &= -\hat{\boldsymbol{\theta}}_k^T\bar{\boldsymbol{\xi}}_k, \quad \hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \gamma N_k\bar{\boldsymbol{\xi}}_{k-1}\frac{\epsilon_k}{d_k}, \quad \hat{\boldsymbol{\theta}}_0 = \epsilon_0 I, \quad \gamma > 0, \\ \epsilon_k &= \frac{\gamma x_k}{g_k}, \quad d_k = 1 + \|\bar{\boldsymbol{\xi}}_{k-1}\|^2 + |N_k| + \epsilon_k^2, \quad g_k = 1 + |N_k|, \end{aligned} \tag{3.20}$$

where N_k is the discrete Nussbaum gain defined to be

$$N_k = z_{s,k}S_{N,k}, \quad z_k = z_{k-1} + \frac{g_{k-1}\epsilon_{k-1}^2}{d_{k-1}}, \quad z_0 = 0, \quad z_{s,k} = \sup_{k' \le k}\{z_{k'}\}, \tag{3.21}$$

and $S_{N,k}$ is the sign function of the discrete Nussbaum gain, i.e., $S_{N,k} = \pm 1$.

3.3.2 Extension to more general nonlinear plants

In most of discrete-time adaptive control works the nonlinearities are restricted to sector-bounded or Lipschitz continuous, i.e., satisfying the linear growth condition. In [65], a least-squares estimator with nonlinear data weighting is developed and used to adaptively control a discrete-time nonlinear system with constant unknown parameters,

where the nonlinear regressor does not satisfy the linear growth condition. Using the lifting technique, this result can be extended to plants with periodic parameters. Consider the plant (3.5) that is the lifted version of the plant (3.1)

$$x_{k+1} = \boldsymbol{\theta}^T \boldsymbol{\xi}_k + u_k,$$

but here we only assume the regressor $\boldsymbol{\xi}_k$ bounded for bounded x_k. By combining a certainty-equivalence controller with a least-squares estimator as below

$$\begin{aligned} u_k &= -\hat{\boldsymbol{\theta}}_k^T \boldsymbol{\xi}_k, \;\; \hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \left(1 + \boldsymbol{\xi}_k^T \boldsymbol{\xi}_k\right) e_k P_k \boldsymbol{\xi}_{k-1}, \;\; \hat{\boldsymbol{\theta}}_0 = \epsilon_0 I, \\ e_k &= x_k - u_{k-1} - \hat{\boldsymbol{\theta}}_{k-1}^T \boldsymbol{\xi}_{k-1}, \\ P_k &= P_{k-1} - \frac{\left(1 + \boldsymbol{\xi}_k^T \boldsymbol{\xi}_k\right) P_{k-1} \boldsymbol{\xi}_{k-1} \boldsymbol{\xi}_{k-1}^T P_{k-1}}{1 + \left(1 + \boldsymbol{\xi}_k^T \boldsymbol{\xi}_k\right) \boldsymbol{\xi}_{k-1}^T P_{k-1} \boldsymbol{\xi}_{k-1}}, \quad P_0 = P_0^T > 0, \end{aligned} \tag{3.22}$$

a global stability result can be derived using the same Lyapunov analysis given in [65]. On the contrary, this extension does not hold for the PAC method [1] that employs direct periodic updating. The difficulty to extend the Lyapunov analysis [65] from time-based adaptation to period-based adaptation is due to the mixed time and period based operations in the latter, which is far more complicated than the former.

Next we consider extension to systems with convex/concave parameterizations. For simplicity consider a class of nonlinear discrete-time systems with a single periodic parameter with the periodicity $\mathcal{N}$.

$$x_{k+1} = f(x_k, \theta_k^o) + u_k, \tag{3.23}$$

where $f(\cdot)$ is either a convex or concave function with respect to the argument θ_k^o. In [80], an adaptive control method was proposed to deal with this problem when unknown parameters are time invariant. Now, using the lifting technique, we can write two augmented vectors $\mathbf{f}_k = [f(x_k, \theta_1), \cdots, f(x_k, \theta_{\mathcal{N}})]^T$, $\boldsymbol{\zeta}_k = [0, \cdots, 0, 1, 0, \cdots, 0]^T \in \mathcal{R}^{\mathcal{N}}$, where

θ_i for $i = 1, \cdots, \mathcal{N}$ are constant unknown parameters, and the unity element appears at the jth position only if $k = s\mathcal{N} + j$ and rotates. As such, we have $f(x_k, \theta_k) = \mathbf{f}_k^T \boldsymbol{\zeta}_k$, and can rewrite (3.23) as

$$x_{k+1} = \mathbf{f}_k^T \boldsymbol{\zeta}_k + u_k, \tag{3.24}$$

which contains time-invariant parameters only, hence the design [80] can be applied with minor changes.

3.3.3 Extension to tracking tasks

Consider the plant (3.11) with multiple unknown parameters and the unknown periodic input gain. It is required that the state x_k follows a given reference trajectory $r(k)$. Specifying the tracking error as $e_k = x_k - r_k$, we have

$$e_{k+1} = x_{k+1} - r_{k+1} = \boldsymbol{\theta}_k^{oT} \boldsymbol{\xi}_k + b_k^o u_k - r_{k+1}. \tag{3.25}$$

Using the same lifting technique applied in (3.11), (3.25) can be rewritten in the following form

$$e_{k+1} = \bar{\boldsymbol{\theta}}^T \bar{\boldsymbol{\xi}}_k + \mathbf{b}^T \boldsymbol{\zeta}_k u_k - r_{k+1}. \tag{3.26}$$

To accommodate the tracking task, the adaptive control law (3.31) can be revised as

$$u_k = \frac{1}{\hat{\mathbf{b}}_k^T \boldsymbol{\zeta}_k} \left(r_{k+1} - \hat{\boldsymbol{\theta}}_k^T \bar{\boldsymbol{\xi}}_k \right), \quad \hat{\boldsymbol{\phi}}_k = L \left[\hat{\boldsymbol{\phi}}_{k-1} + \frac{e_k \boldsymbol{\xi}_{k-1}^*}{c + \|\boldsymbol{\xi}_{k-1}^*\|^2} \right], \; c > 0, \; \hat{\boldsymbol{\phi}}(0) = \epsilon_0 I, \tag{3.27}$$

where $\hat{\boldsymbol{\phi}}_k = \left[\hat{\boldsymbol{\theta}}_k^T, \; \hat{\mathbf{b}}_k^T \right]^T$ and $\boldsymbol{\xi}_k^* = \left[\bar{\boldsymbol{\xi}}_k^T, \frac{r_{k+1} - \hat{\boldsymbol{\theta}}_k^T \bar{\boldsymbol{\xi}}_k}{\hat{\mathbf{b}}_k^T \boldsymbol{\zeta}_k} \boldsymbol{\zeta}_k^T \right]^T$. The closed-loop system is then given by

$$e_{k+1} = \tilde{\boldsymbol{\phi}}_k^T \boldsymbol{\xi}_k^*. \tag{3.28}$$

The convergence analysis can be conducted analogous to the tracking problem discussed in [1] by choosing the periodicity $\mathcal{N} = 1$. What differs from [1] is that the adaptation law of (3.27) is a parallel updating.

3.4 Extension to Higher Order Systems

In the preceding extensions, all the plants are kept in one dimensional case. Next, we consider more extensions to higher order systems, where the plant takes the canonical form or the parametric-strict-feedback form.

3.4.1 Extension to canonical systems

First consider the single input higher order system in canonical form

$$\begin{cases} x_{1,k+1} &= x_{2,k}, \\ &\vdots \\ x_{n-1,k+1} &= x_{n,k}, \qquad\qquad \mathbf{x}(0) = \mathbf{x}_0, \\ x_{n,k+1} &= (\boldsymbol{\theta}_k^o)^T \boldsymbol{\xi}_k + b_k^o u_k, \\ y_k &= x_{1,k}, \end{cases} \tag{3.29}$$

where $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T \in \mathcal{R}^n, u_k$, and y_k represent the system states, input and output, respectively. $\boldsymbol{\theta}_k^o = [\theta_{1,k}^o, \cdots, \theta_{m,k}^o]^T \in \mathcal{R}^m$ are unknown periodic parameters, $\boldsymbol{\xi}_k = \boldsymbol{\xi}(\mathbf{x}_k) = [\xi_{1,k}, \cdots, \xi_{m,k}]^T \in \mathcal{R}^m$ is a known vector-valued regressor which is sector bounded, $\|\boldsymbol{\xi}\| \leq c_1 + c_2\|\mathbf{x}\|$ (c_1 and c_2 being arbitrary positive constants), and $b_k^o \in C[0, \infty)$ is a periodic uncertain gain of the system input. To avoid control singularity, it is assumed that b_k^o has a lower bound, that is, $b_k^o \geq b_{min}$ where $b_{min} > 0$ is known. The control objective is to design an adaptive control law consisting of parameter estimation laws $\hat{\boldsymbol{\theta}}_k^o$, $\hat{b}_k^o$ and a control law $u(k, \hat{\boldsymbol{\theta}}_k^o, \hat{b}_k^o)$ such that the system output y_k can follow a desired trajectory r_k in the presence of the parametric uncertainties $\boldsymbol{\theta}_k^o$ and b_k^o.

To derive the lifting adaptive control law, we define the augmented parametric vector

and corresponding vector-valued nonlinearity regressor as in *Extension A* of Section III, by noticing that the n-th subsystem in (3.29) is similar to the plant (3.11). Consequently the plant (3.29) can be rewritten as below

$$\begin{cases} x_{1,k+1} &= x_{2,k}, \\ &\vdots \\ x_{n-1,k+1} &= x_{n,k}, \qquad \mathbf{x}(0) = \mathbf{x}_0, \\ x_{n,k+1} &= \boldsymbol{\theta}^T \bar{\boldsymbol{\xi}}_k + \mathbf{b}^T \boldsymbol{\zeta}_k u_k, \\ y_k &= x_{1,k}, \end{cases} \tag{3.30}$$

where all parametric uncertainties in (3.30) are time invariant.

Now we are in a position to apply existing adaptive control methods. Define $\hat{\boldsymbol{\phi}}_k = [\hat{\boldsymbol{\theta}}_k^T,\ \hat{\mathbf{b}}_k^T]^T$ be the estimation of time-invariant parametric uncertainties $\boldsymbol{\theta}$ and $\mathbf{b}$, $\tilde{\boldsymbol{\phi}}_k = [\tilde{\boldsymbol{\theta}}_k^T,\ \tilde{\mathbf{b}}_k^T]^T$ be the estimation errors where $\tilde{\boldsymbol{\theta}}_k = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k$, $\tilde{\mathbf{b}}_k = \mathbf{b} - \hat{\mathbf{b}}_k$, and $\boldsymbol{\xi}_k^* = \left[\bar{\boldsymbol{\xi}}_k^T, \frac{r_{k+n} - \hat{\boldsymbol{\theta}}_k^T \bar{\boldsymbol{\xi}}_k}{\hat{\mathbf{b}}_k^T \boldsymbol{\zeta}_k} \boldsymbol{\zeta}_k^T\right]^T$. Assuming that all the states are available, the following adaptive control law is proposed

$$u_k = \frac{1}{\hat{\mathbf{b}}_k^T \boldsymbol{\zeta}_k} \left(r_{k+n} - \hat{\boldsymbol{\theta}}_k^T \bar{\boldsymbol{\xi}}_k \right), \tag{3.31}$$

$$\hat{\boldsymbol{\phi}}_k = L \left[\hat{\boldsymbol{\phi}}_{k-1} + P_k \boldsymbol{\xi}_{k-1}^* e_{n,k} \right], \quad \hat{\boldsymbol{\phi}}_0 = \epsilon_0 I, \tag{3.32}$$

$$P_k = P_{k-1} - \frac{P_{k-1} \left(\boldsymbol{\xi}_{k-1}^*\right)^T \boldsymbol{\xi}_{k-1}^* P_{k-1}}{1 + \left(\boldsymbol{\xi}_{k-1}^*\right)^T P_{k-1} \boldsymbol{\xi}_{k-1}^*}, \ P_0 = P_0^T > 0, \tag{3.33}$$

where $e_{n,k} = x_{n,k} - r_{k+n-1}$, and the covariance P_k is a positive definite matrix derived from the relationship $P_k^{-1} = P_k^{-1} + \boldsymbol{\xi}_{k-1}^* \left(\boldsymbol{\xi}_{k-1}^*\right)^T$. Note that the parameter estimate (3.32) is dependent on $x_{n,k}$ where the subscript n denotes the n-th state variable. Define the tracking error as $\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}_{r,k} = [\mathbf{e}_{a,k}^T\ e_{n,k}]^T$, where $\mathbf{x}_{r,k} = [r_k, r_{k+1}, \cdots, r_{k+n-1}]^T$. Substituting the control (3.31) into (3.30), the closed-loop system is

$$\mathbf{e}_{a,k+1} = \begin{bmatrix} \mathbf{0} & I_{n-2} \\ 0 & \mathbf{0} \end{bmatrix} \mathbf{e}_{a,k} + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} e_{n,k}, \tag{3.34}$$

$$e_{n,k+1} = \tilde{\boldsymbol{\phi}}_k^T \boldsymbol{\xi}_k^*. \tag{3.35}$$

Theorem 3.2 *Under the lifted adaptation law (3.32) and (3.33), the closed-loop dynamics (3.34) and (3.35) are asymptotically stable.*

Proof. The convergence analysis can also be conducted analogous to the PAC [1] by choosing the periodicity $\mathcal{N} = 1$. ■

3.4.2 Extension to parametric-strict-feedback systems

Consider the adaptive control of parametric-strict-feedback systems with periodic uncertainties, described by

$$\begin{aligned} x_{1,k+1} &= (\boldsymbol{\theta}^o_{1,k})^T \boldsymbol{\xi}_{1,k} + b^o_{1,k} x_{2,k}, \\ &\vdots \\ x_{n-1,k+1} &= (\boldsymbol{\theta}^o_{n-1,k})^T \boldsymbol{\xi}_{n-1,k} + b^o_{n-1,k} x_{n,k}, \\ x_{n,k+1} &= (\boldsymbol{\theta}^o_{n,k})^T \boldsymbol{\xi}_{n,k} + b^o_{n,k} u_k, \\ y_k &= x_{1,k}, \end{aligned} \tag{3.36}$$

where $\mathbf{x}_l = [x_1, x_2, \cdots, x_l]^T$ are system states, y_k is the system output, $\boldsymbol{\theta}^o_{l,k} \in \mathcal{R}^{m_l}$, $b^o_{l,k} \in \mathcal{R}, l = 1, \cdots, n$, are unknown periodic parameters (m_l's are positive integers), $\boldsymbol{\xi}_{l,k} \triangleq \boldsymbol{\xi}_l(\mathbf{x}_{l,k}) \in \mathcal{R}^{m_l}$ denotes the known nonlinear regressor which is Lipschitz continuous. Assume that each control gain $b^o_{l,k}$ always takes a positive or negative sign for all k, the periodicity of the i-th element of $\boldsymbol{\theta}^o_l$ or $\theta^o_{i,l}$ is $\mathcal{N}_{i,l}$ for $i = 1, \cdots, m_l$ and $l = 1, \cdots, n$, and the periodicity of b^o_l is $\mathcal{N}_{l,b}$ for $l = 1, \cdots, n$. Without loss of generality, assume $\mathcal{N}_{i,l}$ and $\mathcal{N}_{l,b}$ are all relatively prime. The control objective is to make the output y_k track a bounded reference trajectory r_k and at the same time guarantee the boundedness of all the closed-loop signals.

When the control gains $b^o_{l,k} = 1$ and $\boldsymbol{\theta}^o_{l,k}$ are constant unknowns, the system (3.36) is in the parametric-strict-feedback form and has been studied in [166], [169]- [174]. When all the unknown parameters in (3.36) are constant, a robust adaptive control scheme is proposed [39], which uses a future state predictor and a discrete Nussbaum gain. When

the control directions are known *a priori*, [45] a PAC scheme is proposed in which the least common multiple is used as the sole updating period. It is an open issue when the control directions are uncertainties and periodic. In the following, by applying the lifting technique the adaptive control [39] is extended to (3.36).

Rewrite the plant (3.36) as

$$\begin{aligned} y_{k+n} &= (\boldsymbol{\theta}^o_{1,k+n-1})^T \boldsymbol{\xi}_{1,k+n-1} + b^o_{1,k+n-1} x_{2,k+n-1}, \\ x_{l,k+n-l+1} &= (\boldsymbol{\theta}^o_{l,k+n-l})^T \boldsymbol{\xi}_{l,k+n-l} + b^o_{l,k+n-l} x_{l,k+n-l}, \\ x_{n,k+1} &= (\boldsymbol{\theta}^o_{n,k})^T \boldsymbol{\xi}_{n,k} + b^o_{n,k} u_k, \end{aligned} \tag{3.37}$$

with $l = 2, \cdots, n-1$. By iterative substitution, the output equations can be written as

$$\begin{aligned} y_{k+n} &= \sum_{l=1}^{n} \prod_{i=1}^{l-1} b^o_{i,k+n-i} (\boldsymbol{\theta}^o_{l,k+n-l})^T \boldsymbol{\xi}_{l,k+n-l} \\ &\quad + \prod_{i=1}^{n} b^o_{i,k+n-i} u_k. \end{aligned} \tag{3.38}$$

Define

$$\begin{aligned} \Theta^o_k &= \Big[(\boldsymbol{\theta}^o_{1,k+n-1})^T, b^o_{1,k+n-1} (\boldsymbol{\theta}^o_{2,k+n-2})^T, \cdots, \\ &\quad \prod_{i=1}^{n-1} b^o_{i,k+n-i} (\boldsymbol{\theta}^o_{n,k})^T \Big]^T \in \mathcal{R}^{\sum_{l=1}^{n} m_l}, \\ \Psi^o_{k+n-1} &= [\boldsymbol{\xi}^T_{1,k+n-1}, \cdots, \boldsymbol{\xi}^T_{n,k}]^T \in \mathcal{R}^{\sum_{l=1}^{n} m_l}, \\ g^o_k &= \prod_{i=1}^{n} b^o_{i,k+n-i} \in \mathcal{R}, \end{aligned}$$

the output equation (3.38) can be further written in a compact form as

$$y_{k+n} = (\Theta^o_k)^T \Psi^o_{k+n-1} + g^o_k u_k. \tag{3.39}$$

Note that Θ^o_k and g^o_k are still periodic unknowns.

Future state prediction

In (3.39), a key issue is the noncausal problem due because functions Ψ^o_{k+n-1} depend on the furture states. It is noted from (3.36) that the future states $\mathbf{x}_{l,k+n-l}, l =$

$1, 2, \cdots, n-1$, are deterministic at the k-th step because they are independent of control input u_k.

Comparing (3.36) with (3.11), we can see that every x_l subsystem in (3.36) takes a form analogous to (3.11). Using the lifting technique, $\boldsymbol{\theta}_{l,k}^o$ can be converted into an augmented vector $\boldsymbol{\theta}_l \in \mathcal{R}^{\mathcal{N}_l}$ that is exactly the same as that derived in (3.12), where $\mathcal{N}_l = \sum_{i=1}^{m_l} \mathcal{N}_{i,l}$. Likewise, note that similarity between $\boldsymbol{\xi}_{l,k}$ in (3.36) and $\boldsymbol{\xi}_k$ in (3.11), $\boldsymbol{\xi}_{l,k}$ can be converted to an augmented vector-valued regressor analogous to $\bar{\boldsymbol{\xi}}_k$ as we did for $\boldsymbol{\xi}_k$. Referring to (3.13), we have

$$\bar{\boldsymbol{\xi}}_{l,k} = \left[\bar{\boldsymbol{\xi}}_{l,1,k}^T, \cdots, \bar{\boldsymbol{\xi}}_{l,m_l,k}^T\right]^T \in \mathcal{R}^{\mathcal{N}_l}, \tag{3.40}$$

where $\bar{\boldsymbol{\xi}}_{l,i,k} = [0, \cdots, 0, \xi_{l,i,k}, 0, \cdots, 0]^T \in \mathcal{R}^{N_{i,l}}$, and the element $\xi_{l,i,k}$ appears in the jth position of $\bar{\boldsymbol{\xi}}_{l,i,k}$ only if $k = s\mathcal{N}_{i,l} + j$, for $j = 1, 2, \cdots, \mathcal{N}_{i,l}$. Here the first index in the subscript of $\xi_{l,i,k}$ denotes the lth subsystem in (3.36), the second index in the subscript of $\xi_{l,i,k}$ denotes the ith element of $\boldsymbol{\xi}_{l,k}$, and the third index in the subscript of $\xi_{l,i,k}$ denotes the time instance k.

Similarly we can convert $b_{l,k}^o$ to $\mathbf{b}_l^T \boldsymbol{\zeta}_{l,k}$, with $\mathbf{b}_l = [b_{l,1}, \cdots, b_{l,\mathcal{N}_{l,b}}]^T$ being time invariant, and $\boldsymbol{\zeta}_{l,k} = [0, \cdots, 0, 1, 0, \cdots, 0]^T \in \mathcal{R}^{\mathcal{N}_{l,b}}$, where the only non-zero element, 1, appears in the jth position of $\boldsymbol{\zeta}_{l,k}$ when $k = s\mathcal{N}_{l,b} + j$.

Subsequently, the first $n-1$ state equations in (3.36) are equivalent to

$$x_{l,k+1} \;=\; \boldsymbol{\theta}_l^T \bar{\boldsymbol{\xi}}_{l,k} + \mathbf{b}_l^T \bar{\mathbf{x}}_{l+1,k}, \quad l = 1, \cdots, n-1, \tag{3.41}$$

where $\bar{\mathbf{x}}_{l+1,k} = \boldsymbol{\zeta}_{l,k} x_{l+1,k} \in \mathcal{R}^{\mathcal{N}_{l,b}}$, and all the unknowns are time invariant. Then, by extending the result in [39] from scalar unknown case to vector unknown case, a state predictor can be derived. The derivation procedure is shown in *Appendix A.2*.

Controller design

In designing the controller for the plant (3.39), two subcases are taken into consideration (i) the control directions are known, and (ii) the control directions are unknown.

In (3.39), using the lifting technique, Θ_k^o can be converted into an augmented constant vector $\Theta \in \mathcal{R}^\kappa, \kappa \triangleq \sum_{l=1}^{n} \prod_{j=1}^{l-1} \mathcal{N}_{j,b}\mathcal{N}_l$, $\mathcal{N}_l = \sum_{i=1}^{m_l} \mathcal{N}_{i,l}$, and the associated $\Psi_{k+n-1} = [\Psi_1^T(\mathbf{x}_1(k+n-1)),\ \Psi_2^T(\mathbf{x}_2(k+n-2)), \cdots, \Psi_n^T(\mathbf{x}_n(k))]^T \in \mathcal{R}^\kappa$ is an extended regressor, satisfying

$$\left(\Theta_{k+n-1}^o\right)^T \Psi_{k+n-1}^o = \Theta^T \Psi_{k+n-1}. \tag{3.42}$$

Thus, (3.39) is equivalent to

$$y_{k+n} = \Theta^T \Psi_{k+n-1} + g_k^o u_k. \tag{3.43}$$

Similarly, it is convenient to convert the control gain g_k^o to $\mathbf{g}^T\boldsymbol{\zeta}_k$, with $\mathbf{g} \in \mathcal{R}^{\kappa_b}, \kappa_b \triangleq \prod_{j=1}^{n} \mathcal{N}_{j,b}$ being time invariant, and $\boldsymbol{\zeta}_k = [0, \cdots, 0, 1, 0, \cdots, 0]^T \in \mathcal{R}^{\kappa_b}$, where the only non-zero element, 1, appears in the jth position of $\boldsymbol{\zeta}_k$ when $k = s\kappa_b + j$. Thus, (3.43) can be further expressed as

$$y_{k+n} = \Theta^T \Psi_{k+n-1} + \mathbf{g}^T\boldsymbol{\zeta}_k u_k. \tag{3.44}$$

Case (i): The control directions are known.

Since (3.43) is transformed to a system with unknown time-invariant parameters Θ and $\mathbf{g}$, we can design a discrete-time adaptive backstepping control law so far as the control directions are known *a priori*. Define $\hat{\boldsymbol{\phi}}_k = [\hat{\Theta}_k^T,\ \hat{\mathbf{g}}_k^T]^T$ be the estimation of time-invariant parametric uncertainties Θ and $\mathbf{g}$, and

$$\Psi_{k+n-1}^* = \left[\Psi_{k+n-1}^T, \frac{r_{k+n} - \hat{\Theta}_k^T \hat{\Psi}(k+n-1|k)}{\hat{\mathbf{g}}_k^T \boldsymbol{\zeta}_k} \boldsymbol{\zeta}_k^T\right]^T.$$

The following adaptive control law is proposed

$$
\begin{aligned}
u_k &= \frac{1}{\hat{\mathbf{g}}_k^T \boldsymbol{\zeta}_k} \left(r_{k+n} - \hat{\Theta}_k^T \hat{\Psi}(k+n-1|k) \right), && (3.45)\\
\hat{\boldsymbol{\phi}}_k &= L \left[\hat{\boldsymbol{\phi}}_{k-n} + P_k \Psi_{k-1}^* e_k \right], && (3.46)\\
\hat{\boldsymbol{\phi}}_j &= \epsilon_0 I, \quad j = 0, -1, \cdots, -n+1, \\
P_k &= P_{k-n} - \frac{P_{k-n} \left(\Psi_{k-1}^* \right)^T \Psi_{k-1}^* P_{k-n}}{1 + \left(\Psi_{k-1}^* \right)^T P_{k-n} \Psi_{k-1}^*}, && (3.47)
\end{aligned}
$$

where $P_j = P_j^T > 0$, $e_k = y_k - r_k$, and $\hat{\Psi}(k+n-1|k)$ is the prediction of Ψ_{k+n-1} at step k.

Theorem 3.3 *Consider the adaptive closed-loop system consisting of system (3.36), adaptive control (3.45) with the corresponding parameter updating law (3.46) and (3.47), future state prediction (8.55)–(8.18) and parameter estimation law (8.19). All the signals in the closed-loop system are guaranteed to be bounded and the tracking error e_k converges to zero asymptotically.*

Proof. Substituting the adaptive control (3.45) into the n-step predictor (3.43) and subtracting r_{k+n} on both hand sides, we obtain the following error dynamics

$$
e_{k+n} = \tilde{\boldsymbol{\phi}}_k^T \Psi_{k+n-1}^* + \hat{\Theta}_k^T (\Psi_{k+n-1} - \hat{\Psi}(k+n-1|k)) \tag{3.48}
$$

with $\tilde{\boldsymbol{\phi}}_k = [\tilde{\Theta}_k^T,\ \tilde{\mathbf{g}}_k^T]^T$, where $\tilde{\Theta}_k = \Theta - \hat{\Theta}_k$ and $\tilde{\mathbf{g}}_k = \mathbf{g} - \hat{\mathbf{g}}_k$. Select a nonnegative function $V_k = \tilde{\boldsymbol{\phi}}_k^T P_k^{-1} \tilde{\boldsymbol{\phi}}_k$ and consider its difference with respect to n

$$
\begin{aligned}
\Delta V_k &= V_k - V_{k-n} \\
&\leq \tilde{\boldsymbol{\phi}}_{k-n}^T (\Psi_{k-1}^* (\Psi_{k-1}^*)^T) \tilde{\boldsymbol{\phi}}_{k-n} - 2 \tilde{\boldsymbol{\phi}}_{k-n}^T \Psi_{k-1}^* e_k \\
&\quad + (\Psi_{k-1}^*)^T P_k \Psi_{k-1}^* e_k^2 \\
&= -e_k^2 + (\hat{\Theta}_{k-n}^T (\Psi_{k-1} - \hat{\Psi}(k-1|k-n)))^2
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{(\Psi_{k-1}^*)^T P_{k-n}\Psi_{k-1}^*}{1+(\Psi_{k-1}^*)^T P_{k-n}\Psi_{k-1}^*}e_k^2\\
= &-\frac{e_k^2}{1+(\Psi_{k-1}^*)^T P_{k-n}\Psi_{k-1}^*}\\
&+(\hat{\Theta}_{k-n}^T(\Psi_{k-1}-\hat{\Psi}(k-1|k-n)))^2. \qquad (3.49)
\end{aligned}
$$

Due to the linear growth condition of nonlinear regressors in system (3.36) and the boundedness of reference signal r_k, two properties with respect to the augmented regressor Ψ_{k+n-1} in (3.43) can be obtained from *Lemma* 2 and *Lemma* 7 in [39] as follows.

a) $\Psi_{k+n-1}=O[e_{k+n-1}]$, namely, there exist positive constants c_1, c_2 and k_0 such that $\|\Psi_{k+n-1}\|\le c_1\max_{i\le k+n-1}|e_i|+c_2$, for all $k\ge k_0$;

b) $\Psi_{k+n-1}-\hat{\Psi}(k+n-1|k)=o[e_{k+n-1}]$, namely, there exists a sequence satisfying $\lim_{i\to\infty}\alpha_i\to 0$ and a constant k_0 such that $\|\Psi_{k+n-1}-\hat{\Psi}(k+n-1|k)\|=\alpha_{k+n-1}\max_{i\le k+n-1}|e_i|$, for all $k\ge k_0$.

From properties a) and b), it is easy to see that $\Psi_{k-1}^*=O[e_{k-1}]=O[e_k]$. Subsequently, the parametric adaptation law (3.46)–(3.47) induces that $\hat{\boldsymbol{\phi}}_k=O[1]$, or furthermore

$$\hat{\Theta}_{k-n}^T(\Psi_{k-1}-\hat{\Psi}(k-1|k-n))=o[e_{k-1}]=o[e_k].$$

From (3.49), the nonnegative property of V_k implies that

$$\frac{e_k^2}{1+(\Psi_{k-1}^*)^T P_{k-n}\Psi_{k-1}^*}\to 0. \qquad (3.50)$$

Considering the Key Technical Lemma under the sector condition $\|\Psi_{k-1}^*\|\le c_1\max_{i\le k}|e_i|+c_2$, $e_k\to 0$. The proof is complete. ■

Case (ii): The control directions are unknown.

In this case, the adaptive control design is a mixture of the lifting and periodic updating algorithms. The classic Nussbaum gain method [74] is limited to time-invariant *scalar*

control gains. As we can see from the above reformulation with the lifting approach, the periodic input gain g_k^o becomes the product of two vectors $\mathbf{g}$ and $\boldsymbol{\zeta}_k$.

To overcome this difficulty, we adopt periodic adaptations to deal with the periodic input gains, meanwhile apply the lifting technique to convert the remaining periodic parameters, $\boldsymbol{\theta}_{l,k}^o$, into time-invariant Θ, as shown in (3.44). Note that κ_b is the periodicity of gain function g_k^o. Thus the periodic updating mechanism for g_k^o consists of κ_b elements, each is a constant with respect to the periodicity κ_b. Consequently the application of Nussbaum gain becomes feasible. The proposed controller for this case is a combination of the lifting technique and periodic adaptation, which is given in *Appendix A.3*.

From the dynamics $y_{k+n} = \Theta^T \Psi_{k+n-1} + g_k^o u_k$ in (3.43), one can derive the ideal input

$$u_k = -\left(g_k^o\right)^{-1} \Theta^T \Psi_{k+n-1} + \left(g_k^o\right)^{-1} r_{k+n}. \tag{3.51}$$

Referring to (3.13) and (3.40), the lifting technique gives

$$u_k = -\Theta_g^T \bar{\Psi}_{k+n-1} + \mathbf{g}_0^T \mathbf{r}_{k+n}, \tag{3.52}$$

where $\Theta_g \in \mathcal{R}^{\kappa\kappa_b}$ and $\mathbf{g}_0 \in \mathcal{R}^{\kappa_b}$ are time-invariant unknown vectors, $\bar{\Psi}_{k+n-1}$ and $\mathbf{r}_{k+n} \stackrel{\triangle}{=} \boldsymbol{\zeta}_k r_{k+n}$ are the corresponding regressors, satisfying

$$\frac{1}{\mathbf{g}^T \boldsymbol{\zeta}_k} \Theta^T \Psi_{k+n-1} = \Theta_g^T \bar{\Psi}_{k+n-1}, \ \frac{1}{\mathbf{g}^T \boldsymbol{\zeta}_k} r_{k+n} = \mathbf{g}_0^T \mathbf{r}_{k+n}.$$

Considering the prediction of $\bar{\Psi}_{k+n-1}$ at step k, $\hat{\bar{\Psi}}(k+n-1|k)$, and the adaptation law of system uncertainties (8.22) and (8.23), the control law (3.52) can be revised into

$$u_k = -\hat{\Theta}_g^T(k) \hat{\bar{\Psi}}(k+n-1|k) + \hat{\mathbf{g}}_0^T(k) \mathbf{r}_{k+n}. \tag{3.53}$$

Substituting (3.53) into (3.43), the error dynamics is

$$e_{k+n} \ = \ y_{k+n} - r_{k+n}$$

$$\begin{aligned} &= \Theta^T \Psi_{k+n-1} - g_k^o \hat{\Theta}_g^T(k) \hat{\bar{\Psi}}(k+n-1|k) \\ &\quad + g_k^o \hat{\mathbf{g}}_0^T(k) \mathbf{r}_{k+n} - r_{k+n} \\ &= -g_k^o \tilde{\Theta}_g^T(k) \bar{\Psi}_{k+n-1} + g_k^o \tilde{\mathbf{g}}_0^T(k) \mathbf{r}_{k+n} - g_k^o \beta_{k+n-1}, \end{aligned} \tag{3.54}$$

where $\beta_{k+n-1} = \hat{\Theta}_g^T(k) \hat{\bar{\Psi}}(k+n-1|k) \triangleq \hat{\Theta}_g^T(k)(\hat{\bar{\Psi}}(k+n-1|k) - \bar{\Psi}_{k+n-1})$.

Theorem 3.4 *Consider the adaptive closed-loop system consisting of the plant (3.36), adaptive control law (3.53), parameter updating law (8.22)–(8.23), future state prediction (8.55)–to (8.18) with parameter estimation law (8.19). All the signals in the closed-loop system are guaranteed to be bounded and the tracking error e_k converges to zero asymptotically.*

Proof. Substituting the error dynamics (3.54) into the augmented error $\epsilon_{i,s}$ defined in (8.21), one obtains

$$\begin{aligned} &\gamma \tilde{\Theta}_g^T(\kappa_b(s-n)+i) \bar{\Psi}_{\kappa_b(s-n)+n+i-1} \\ &-\gamma \tilde{\mathbf{g}}_0^T(\kappa_b(s-n)+i) \mathbf{r}_{\kappa_b(s-n)+n+i} \\ &= -\frac{1}{g^o_{\kappa_b(s-n)+i}} \epsilon_{i,s} G_{i,s} - \gamma \beta_{i,s} + \frac{1}{g^o_{\kappa_b(s-n)+i}} N_{i,s} \phi_{i,s} \beta_{i,s}, \end{aligned} \tag{3.55}$$

where the quantities $\phi_{i,s}, \beta_{i,s}, N_{i,s}$, and $G_{i,s}$ are given in (8.24), (8.26), (8.27), and (8.28) respectively, and $\gamma > 0$ is an arbitrary constant. Consider positive definite functions $V_{i,s}$, $i = 0, 1, \cdots, \kappa_b - 1$, as below

$$\begin{aligned} V_{i,s} &= \sum_{j=1}^{n} \tilde{\Theta}_g^T(\kappa_b(s-n+j)+i) \tilde{\Theta}_g(\kappa_b(s-n+j)+i) \\ &\quad + \sum_{j=1}^{n} \tilde{\mathbf{g}}_0^T(\kappa_b(s-n+j)+i) \tilde{\mathbf{g}}_0(\kappa_b(s-n+j)+i). \end{aligned}$$

Noticing the parametric estimate (8.22) and (8.23), the difference of $V_{i,s}$ between two consecutive steps in s can be derived below

$$\Delta V_{i,s} = V_{i,s} - V_{i,s-1}$$

$$
\begin{aligned}
&= \Big[\tilde{\Theta}_g^T(\kappa_b s+i)\tilde{\Theta}_g(\kappa_b s+i) \\
&\quad -\tilde{\Theta}_g^T(\kappa_b(s-n)+i)\tilde{\Theta}_g(\kappa_b(s-n)+i)\Big] \\
&\quad +\big[\tilde{\mathbf{g}}_0^T(\kappa_b s+i)\tilde{\mathbf{g}}_0(\kappa_b s+i) \\
&\quad -\tilde{\mathbf{g}}_0^T(\kappa_b(s-n)+i)\tilde{\mathbf{g}}_0(\kappa_b(s-n)+i)\big] \\
&= \Big[\Big(\tilde{\Theta}_g(\kappa_b s+i)-\tilde{\Theta}_g(\kappa_b(s-n)+i)\Big)^T \\
&\quad \times\Big(\tilde{\Theta}_g(\kappa_b s+i)-\tilde{\Theta}_g(\kappa_b(s-n)+i)\Big) \\
&\quad +2\tilde{\Theta}_g^T(\kappa_b(s-n)+i) \\
&\quad \Big(\tilde{\Theta}_g(\kappa_b s+i)-\tilde{\Theta}_g(\kappa_b(s-n)+i)\Big)\Big] \\
&\quad +\Big[\big(\tilde{\mathbf{g}}_0(\kappa_b s+i)-\tilde{\mathbf{g}}_0(\kappa_b(s-n)+i)\big)^T \\
&\quad (\tilde{\mathbf{g}}_0(\kappa_b s+i)-\tilde{\mathbf{g}}_0(\kappa_b(s-n)+i)) \\
&\quad +2\tilde{\mathbf{g}}_0^T(\kappa_b(s-n)+i) \\
&\quad \times(\tilde{\mathbf{g}}_0(\kappa_b s+i)-\tilde{\mathbf{g}}_0(\kappa_b(s-n)+i))\big] \\
&= \gamma^2\frac{N_{i,s}^2}{D_{i,s}^2}\epsilon_{i,s}^2\Big(\bar{\Psi}^T_{\kappa_b(s-n)+n+i-1}\bar{\Psi}_{\kappa_b(s-n)+n+i-1} \\
&\quad +\mathbf{r}^T_{\kappa_b(s-n)+n+i}\mathbf{r}_{\kappa_b(s-n)+n+i}\Big) \\
&\quad +2N_{i,s}\frac{\gamma\tilde{\Theta}_g^T(\kappa_b(s-n)+i)\bar{\Psi}_{\kappa_b(s-n)+n+i-1}}{D_{i,s}}\epsilon_{i,s} \\
&\quad -2N_{i,s}\frac{\gamma\tilde{\mathbf{g}}_0^T(\kappa_b(s-n)+i)\mathbf{r}_{\kappa_b(s-n)+n+i}}{D_{i,s}}\epsilon_{i,s}.
\end{aligned}
$$

Note the relationships [39]

$$\Delta\chi_{i,s}=\Delta z_{i,s}+\phi_{i,s}\Delta\phi_{i,s}+\frac{[\Delta\phi_{i,s}]^2}{2},$$

$$0\le\Delta z_{i,s}\le 1,\qquad 0\le|\Delta\phi_{i,s}|\le 1,$$

$$|N_{i,s}|[\Delta\phi_{i,s}]^2\le\Delta z_{i,s},$$

where $\phi_{i,s}, z_{i,s}, \chi_{i,s}$ are defined in (8.24), (8.25), and (8.27), we have

$$\begin{aligned}
\Delta V_{i,s} &\leq \gamma^2 \frac{G_{i,s}\epsilon_{i,s}^2}{D_{i,s}} - 2\gamma \frac{N_{i,s}\beta_{i,s}\epsilon_{i,s}}{D_{i,s}} \\
&\quad - \frac{2}{g^o_{\kappa_b(s-n)+i}} N_{i,s} \frac{G_{i,s}\epsilon_{i,s}^2}{D_{i,s}} \\
&\quad + \frac{2}{g^o_{\kappa_b(s-n)+i}} N_{i,s} \frac{N_{i,s}\phi_{i,s}\beta_{i,s}\epsilon_{i,s}}{D_{i,s}} \\
&\leq \gamma^2 \Delta z_{i,s} + 2\gamma \Delta\phi_{i,s} - \frac{2}{g^o_{\kappa_b(s-n)+i}} N_{i,s} \\
&\quad \times \left(\Delta z_{i,s} + \phi_{i,s}\Delta\phi_{i,s} + \frac{[\Delta\phi_{i,s}]^2}{2} \right) \\
&\quad + \frac{1}{|g^o_{\kappa_b(s-n)+i}|} |N_{i,s}| [\Delta\phi_{i,s}]^2 \\
&\leq c_1 \Delta z_{i,s} + 2\gamma\Delta\phi_{i,s} - \frac{2}{g^o_{\kappa_b(s-n)+i}} N_{i,s}\Delta\chi_{i,s},
\end{aligned}$$

where $c_1 = \gamma^2 + \frac{1}{|g^o_{\kappa_b(s-n)+i}|}$. Since $g^o_{\kappa_b(s-n)+i}$ is a constant for each i, taking summation with respect to s on both sides of the above inequality yields

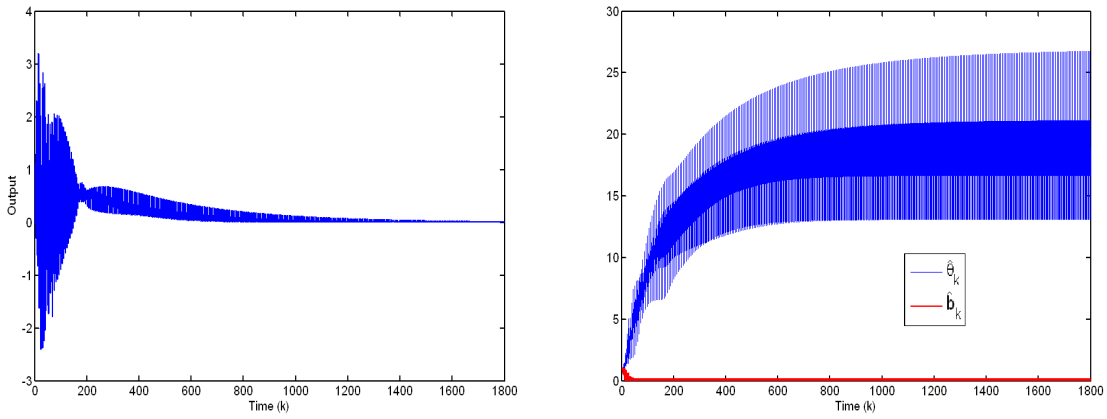
$$\begin{aligned}
V_{i,s} &\leq \sum_{s'=0}^{s} \left[-\frac{2}{g^o_{\kappa_b(s'-n)+i}} N_{i,s'}\Delta\chi_{i,s'} \right] \\
&\quad + c_1 z_{i,s} + c_1 + 2\gamma\phi_{i,s} + 2\gamma + V_{i,-1} \\
&\leq -\frac{2}{g^o_{\kappa_b(s-n)+i}} \sum_{s'=0}^{s} N_{i,s'}\Delta\chi_{i,s'} \\
&\quad + c_1 (z_{i,s} + \frac{\phi_{i,s}^2}{2}) + c_1 + \frac{2\gamma^2}{c_1} + 2\gamma + V_{i,-1} \\
&\leq -\frac{2}{g^o_{\kappa_b(s-n)+i}} \sum_{s'=0}^{s} N_{i,s'}\Delta\chi_{i,s'} + c_1\chi_{i,s} + c_2, \qquad (3.56)
\end{aligned}$$

where $c_2 = c_1 + \frac{2\gamma^2}{c_1} + 2 + V_{i,-1}$. Applying the oscillating-unbounded sum property of discrete Nussbaum gain [74] to (3.56) yields the boundedness of $V_{i,s}$ and $\chi_{i,s}$, and thus the boundedness of $z_{i,s}$ which is a non-decreasing sequence.

The remaining part of proof is similar to that of *Theorem* 1 in [39]. ■

3.5 Illustrative Examples

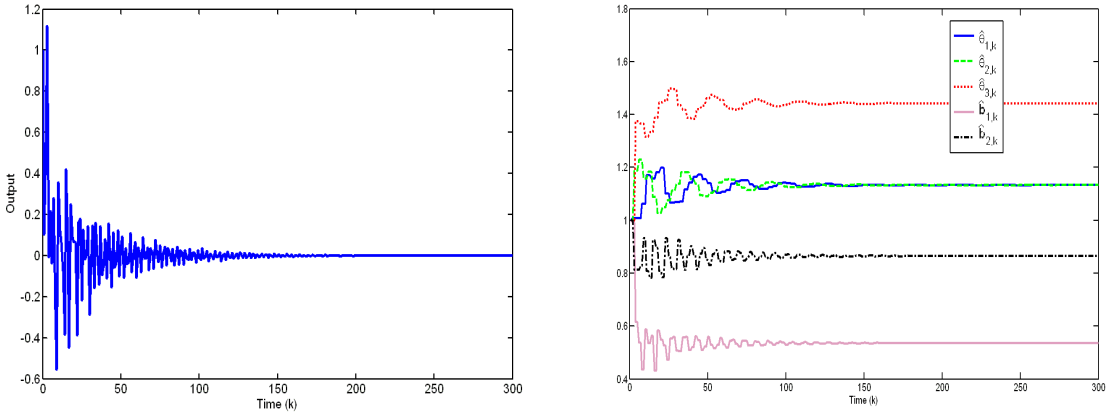
First consider a regulation problem with the dynamics (3.11). Assume $m = 1$, $\theta_k^o = 3.0 + 0.5\cos(2\pi k/3)$, $\xi_k = \cos(x_k + 0.5)$, the input gain $b_k^o = 1.3$ for odd k and $b_k^o = 2.1$ for even k. The known lower bound for b_k^o is 0.1. Figure 3.3 shows the results of the proposed adaptive control with a common period 6 for the parameter updating. A long transient period of adaptation is observed and the error converges after more than 1600 time units. Figure 3.4 shows the results of the proposed method (3.16) and (3.17), where the plant is converted into time-invariant one (3.15) by using the lifting technique. The updating periods are 3 for θ_k^o and 2 for b_k^o, thus there are 5 unknown time-invariant parameters. The results reveal that a much shorter transient response, less than 200 time units, is achieved.



(a) (b)

Figure 3.3: PAC with a common period of 6: (a) regulation error profile; (b) parametric updating profiles.

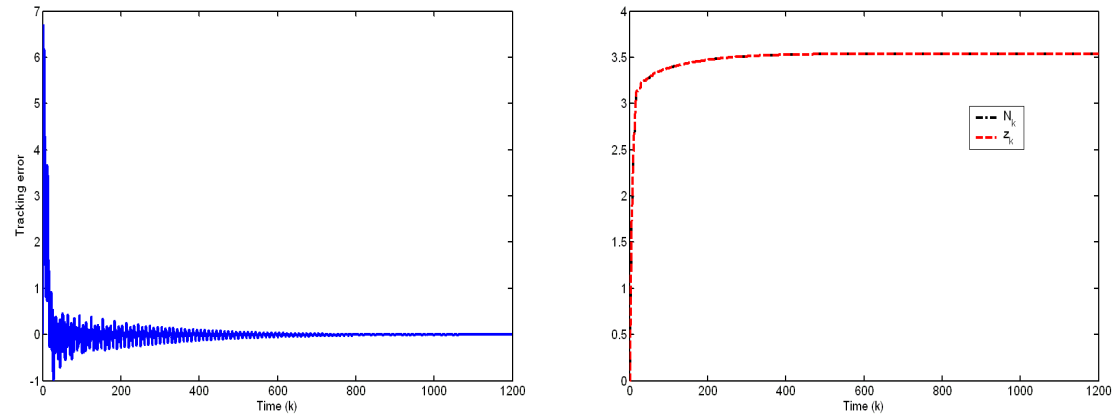
Next consider a trajectory tracking task for the same plant but the control direction is assumed unknown. The given reference is $r_k = 1.6 - 0.8\cos(\pi k/5)$. Apply the discrete Nussbaum gain (3.21) with $\gamma = 1$, and $\epsilon_0 = 2$. The output tracking convergence can



(a) (b)

Figure 3.4: Proposed method using lifting technique: (a) regulation error profile; (b) 5 parametric adaptation profiles.

be achieved, as can be seen from Fig. 3.5(a). Note that since b_k takes positive sign in this example, the discrete Nussbaum gain N_k and its corresponding function z_k always coincide together, as shown in Fig. 3.5(b).

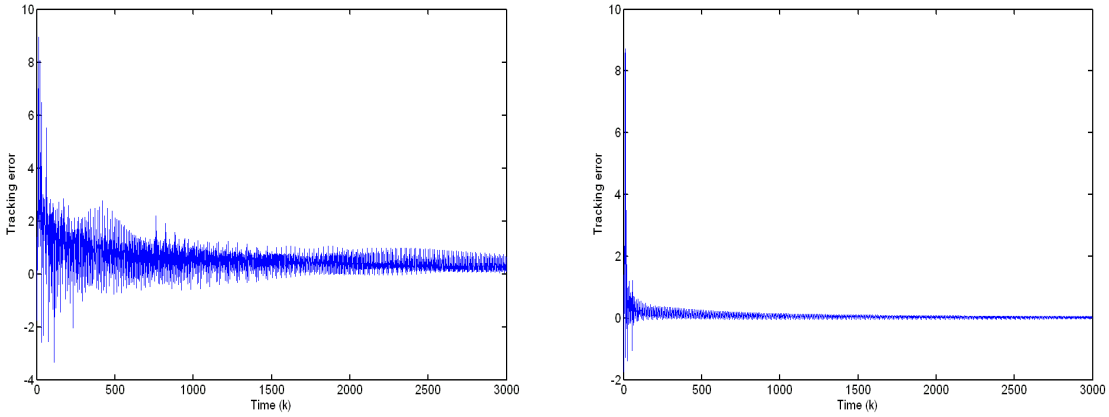


(a) (b)

Figure 3.5: Proposed method using lifting technique and discrete Nussbaum gain: (a) tracking error profile; (b) discrete Nussbaum gain N_k and the corresponding function z_k.

Further consider the same tracking reference and the high-order dynamics (3.29). Assume $n = 3$, $m = 1$, $\mathbf{x}(0) = [0.1\ 0.1\ 0.1]^T$, $\theta_k^o = 3.0 + 0.5\cos(2\pi k/3)$, and $\xi_k = \cos(x_{1,k} + 2x_{3,k} + 0.5)$. The input gain is $b_k^o = 1.3 + 0.5(k - 10[\frac{k-1}{10}] - 2)$, where $[\cdot]$ denotes

the round-down operator. The known lower bound for b_k^o is 0.1. Figure 3.6 shows the results of the PAC with a common period 30 for the parameter updating and the proposed adaptive control with the lifting technique. The proposed method outperforms the PAC.



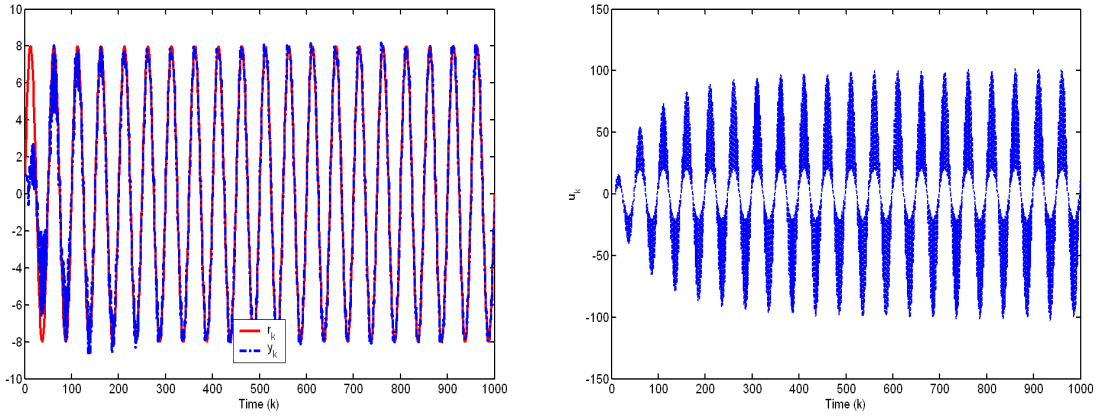
(a) (b)

Figure 3.6: Output tracking error profiles for higher order canonical systems: (a) PAC with a common period of 30; (b) Proposed method using lifting technique.

Finally, consider the following second-order system in parametric-strict-feedback form with unknown control directions

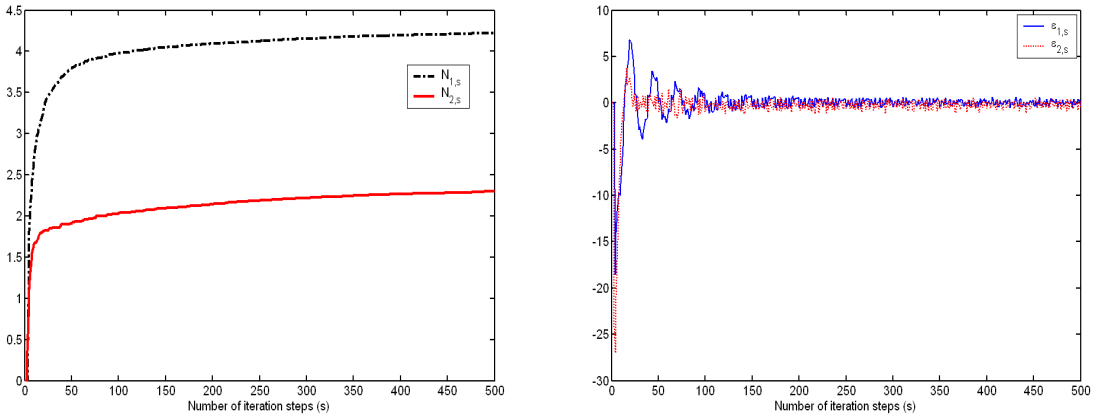
$$\begin{aligned} x_{1,k+1} &= \theta_{1,k}^o \cos x_{1,k} + b_{1,k}^o x_{2,k}, \\ x_{2,k+1} &= \theta_{2,k}^o \sin(x_{1,k} + x_{2,k}) + b_{2,k}^o u_k, \end{aligned}$$

where $\theta_{1,k}^o = -0.3$, $\theta_{2,k}^o = 0.5$, $b_{1,k}^o = 0.8$, $b_{2,k}^o = 0.2$ for odd k and $\theta_{1,k}^o = 0.7$, $\theta_{2,k}^o = -0.5$, $b_{1,k}^o = 0.4$, $b_{2,k}^o = 0.5$ for even k. The tracking reference sequence is $r_k = 8\sin(0.04\pi k)$. The initial system states are $\mathbf{x}_{2,j} = [1,\ 1]^T$, $j = -3, \cdots, 0$. The tuning factor is chosen as $\gamma = 5$. The simulation results are presented in Figs. 3.7 and 3.8. It can be seen that the output tracking achieves a fast convergence even under periodic uncertainties and unknown control directions.



(a) (b)

Figure 3.7: Proposed method for the parametric-strict-feedback system with periodic uncertainties and unknown control directions: (a) output tracking error profile; (b) control input profile.



(a) (b)

Figure 3.8: Proposed method for the parametric-strict-feedback system with periodic uncertainties and unknown control directions: (a) discrete Nussbaum gains; (b) augmented tracking errors $\epsilon_{i,s}$.

3.6 Conclusion

In this chapter we demonstrate that periodic parameters can be converted into time-invariant ones by using a lifting technique. As a result, most existing discrete-time adaptive control methods can be directly applied, even though the original plant is not

satisfying the linear growth condition, nonlinear in parameters, having unknown control directions, or in parametric-strict-feedback form. These issues are open in periodic adaptive control. Using the lifting technique based conversion, we provide a bridge to link control problems characterized by periodic parameters with the existing adaptive control methods, and ultimately provide solutions for these control problems. Further, comparing with the existing periodic adaptive control methods, an immediate advantage of the proposed approach is to expedite the adaptation process greatly owing to the concurrent parametric updating.

In the preceding two chapters, parametric adaptive control schemes are proposed for systems with periodic repetitiveness of uncertainties, in time or spatial domain and in continuous-time or discrete-time space. Actually, besides these, there are many control tasks that are strictly repeatable over finite time intervals. ILC would be efficient to cope with the control of systems with such kind of repetitiveness. In the next four chapters, we shift to those topics.

Chapter 4

Initial State Iterative Learning For Final State Control In Motion Systems

4.1 Introduction

Motion control tasks can be classified into set-point control and tracking control. Set-point control problems arise because of two reasons – only the final states are of concern and specified, and/or the control system is constrained such that only the final states can be controlled. For instance, stopping a moving vehicle at a desired position is a set-point control task. Another example is to shoot a ball into basket, which can only be a set-point control task because it is unnecessary and impossible to specify and control the entire motion trajectory of the ball when only the initial shooting angle and speed are adjustable. Further from energy saving or ecological point of view, we may not want to continuously apply control signals if the desired states can be reached with appropriate initial state values. For instance we can let a train *slip* into and stop at a station with certain initial speed and initial distance. Even if a braking is applied to shorten the slipping time, we may not want to change the braking force so as to keep

a smooth motion for the train. In such circumstances, it is imperative to start slipping from certain appropriate initial position and speed. In this work we focus on final state control of motion systems with initial state manipulation.

The final state control of motion systems can expressed as (1) achieving a desired displacement at a prespecified speed or (2) achieving a desired speed at a prespecified position. It is worth to note the difference between the above two cases. In the first case, imagine that an observer sits in a train and checks the displacement when the train speed drops to a prespecified value. In the second case, imagine that an observer stands in a station and checks the speed when the train enters. In the first case, the information used for control should be the position displacement, whereas in the second case the information used for control should be the speed.

In practice, it is not an easy task to find the appropriate initial states when the desired final states are given, due to two reasons. First, we do not know the exact model of a motion system due to the unknown friction coefficients, unknown load, or other unknown environmental factors such as slope. Thus it is impossible to compute the required initial states as the control inputs. Second, a motion system such as vehicle could be highly nonlinear due to its internal driving characteristics [121] and external interactions with environment in the air, water or on ground such as nonlinear frictions [18]. It is in general impossible to obtain an analytic solution trajectory for such a highly nonlinear dynamics.

On the other hand, many motion control tasks are frequently repeated under the same circumstances, for example the repeated basketball shot exercise, a train entering the same station regularly, an airplane landing on the same runway, etc. The performance of a motion system that executes the same tasks repeatedly can be improved by learning

from previous executions (trials, iterations, passes). Iterative learning control is a suitable method to deal with repeated control tasks [2, 16, 19, 86, 89, 93]. In this chapter, we further demonstrate that ILC is also a suitable method to learn appropriate initial states as control inputs while only the final state information is available.

It should also be noted that the control problems we deal with here are spatial learning tasks. The control specifications are given spatially as the final states where the final states are defined according to a spatial quantity such as the prespecified position or speed, instead of a specific time. In order to build up the direct link between initial and final states and eliminating the time factor, we introduce a state transformation such that motion systems originally described in the time domain become a first order dynamics in the phase plane. Taking one of the initial states as the control input, a simple ILC law updated using one of the final state information can achieve the asymptotic convergence in the iteration axis. Through analysis the convergence conditions for this scenario are made clear.

The chapter is organized as follows. In Section 4.2, some assumptions and induced properties are provided for subsequent sections. In Section 4.3, we focus on initial state iterative learning for final state control, where the system satisfies Lipschitzian condition in position. In Section 4.4, the initial state iterative learning is applied to the motion systems which are Lipschitz continuous in speed. In Section 4.5, initial state iterative learning with optimality is considered when feedback learning control is applied simultaneously. Finally, an illustrative example is provided.

4.2 Problem Formulation and Preliminaries

Consider a motion system

$$\begin{cases} \dfrac{dx}{dt} &= v, \\ \dfrac{dv}{dt} &= -f(x,v), \end{cases} \tag{4.1}$$

where f is continuous on the domain $\mathbf{R}_+^2 \stackrel{\triangle}{=} [0,\infty)\times[0,\infty)$, x is the displacement and v is the speed.

The control objective is to bring the system states (x,v) to an ε-neighbourhood of the desired final state $x_d > 0$ or $v_d \geq 0$ by means of adjusting initial state x_0 or v_0. Clearly the initial states are control inputs. The ε-neighbourhood is defined as $|x_d - x| \leq \varepsilon$ or $|v_d - v| \leq \varepsilon$, where ε is a positive constant. Consider two sets of initial states $x_0 = 0,\ v_0 = u_v$, or $x_0 = u_x,\ v_0 = \mathcal{A}$, where the control inputs u_x and u_v are respectively the initial position and speed, $\mathcal{A}$ is a fixed speed greater than v_d.

In real world most motion systems without control are stable or dissipative in nature. Therefore it is reasonable to assume that a motion system will stop when no exogenous driving control applies. In this work it is assumed that the position $x(t)$ is monotonically increasing, or equivalently

Assumption 4.1 $v \geq 0$.

In motion systems the desired final states may be defined in a very generic manner with the position and speed linked together, that is, defining the final states in the spatial domain. For instance in final position control, the desired final displacement shall be achieved at a prespecified speed, not necessarily at a zero speed. Analogously, in final speed control the desired final speed shall be achieved at a prespecified position.

Now, by eliminating the time t we convert the motion system into the phase plane

(v, x). Dividing the first equation in (4.1) by the second equation yields

$$\frac{dx}{dv} = -g(v, x), \tag{4.2}$$

where $g = v/f$. According to (4.2), the state x is a function of the argument v and control inputs. For simplicity, we write $x(v, u_x)$ when the initial speed is fixed at $\mathcal{A}$ and the control input is the initial position, and write $x(v, u_v)$ when the initial position is fixed at zero and the control input is the initial speed. As far as g is well defined near $f = 0$, the existence and uniqueness of solution ensure that two solution trajectories of (4.1) and (4.2) describe the same physical motion for $v \in [0, \infty)$, one in the time domain and the other in the phase plane. As such, we can derive the same control property when the same control law is applied.

Note that $g(v, x)$ can be viewed as the inverse of generalized damping or friction coefficient. The characteristics of the motion system (4.2) is solely determined by $g(v, x)$.

Assumption 4.2 *For $v,\ x_1,\ x_2 \in \mathbf{R}_+$, there exists a known integrable Lipschitz function $L(v)$ such that*

$$|g(v, x_1) - g(v, x_2)| \leq L(v)|x_1 - x_2|. \tag{4.3}$$

Remark 4.1 *Assumption 4.2 states that the inverse of generalized damping or friction coefficient should meet the Lipschitz continuity condition. In the theory of differential equation, Lipschitz continuity condition is necessary to ensure the existence and uniqueness of the solution trajectory. In motion systems, the solution trajectory should be existing and unique under the same dynamics and same initial condition.*

In practice, many motion systems are discontinuous when speed is zero, due to the

presence of static friction. Consider the Gaussian friction model [18]

$$\begin{cases} \frac{dx}{dt} &= v, \\ \frac{dv}{dt} &= -\frac{1}{m}\left(\left(f_c + (f_s - f_c)e^{-(\frac{v}{v_s})^\delta}\right) sgn(v) + f_v v\right) \end{cases} \tag{4.4}$$

where f_c is the minimum level of kinetic friction, f_s is the level of static friction, f_v is the level of viscous friction, $v_s > 0$ and $\delta > 0$ are empirical parameters. The signum function from static friction represents a non-Lipschitzian term, and owing to this term a vehicle running on ground can always stop in a finite time interval instead of asymptotically stop. The choice of the dx/dv relationship enables the inclusion of the static friction because, according to definition in (4.2), g is continuous both in x and v.

Next define the final position and final speed in spatial domain. In position control, the final displacement, x_e is observed at a prespecified speed v_f. If the initial speed is lower than v_f, v_f cannot be reached. In such circumstance, the final displacement is defined to be

$$x_e(u) \triangleq \begin{cases} x(v,u), & \text{when } v = v_f \\ 0, & v_f \text{ cannot be reached} \end{cases} \tag{4.5}$$

where $x(v_f, u)$ is the position of the system (4.2) at the speed v_f with the control input u.

In speed control, the final speed, v_e, is observed at a prespecified position x_f. However, if the initial speed is low, the final position may not reach x_f while the final speed already drops to zero. In such circumstances, the final speed is defined to be zero. Therefore the final speed is defined in two cases

$$v_e(u) \triangleq \begin{cases} v(x,u), & \text{when } x = x_f \\ 0, & x_f \text{ cannot be reached} \\ & \text{when motion stops.} \end{cases} \tag{4.6}$$

In (4.5) and (4.6), the control input u is either initial position or speed.

From *Assumptions* 4.1 and 4.2, we can derive an important property summarized below.

Property 4.1 *For any two initial quantities $u_j \neq u_j^*$ where (u_j, u_j^*) are either initial positions or speeds, we have $(u_j - u_j^*)[x_e(u_j) - x_e(u_j^*)] > 0$ in final position control and $(u_j - u_j^*)[v_e(u_j) - v_e(u_j^*)] > 0$ in final speed control.*

Proof: See *Appendix A.4*.

4.3 Initial State Iterative Learning

With initial or final position and speed, we have four cases

(i) initial position iterative learning for final position control;

(ii) initial speed iterative learning for final position control;

(iii) initial position iterative learning for final speed control;

(iv) initial speed iterative learning for final speed control.

Denote $x_{i,e}$ and $v_{i,e}$ the final position and speed defined in (4.5) and (4.6) respectively at the ith iteration, where $i = 1, 2, \cdots$ denotes the iteration number. The ILC algorithms corresponding to the four cases are

$$\begin{aligned} &\text{(i)} \;\; u_{x,i+1} = u_{x,i} + \gamma(x_d - x_{i,e}) \\ &\text{(ii)} \;\; u_{v,i+1} = u_{v,i} + \gamma(x_d - x_{i,e}) \\ &\text{(iii)} \;\; u_{x,i+1} = u_{x,i} + \gamma(v_d - v_{i,e}) \\ &\text{(iv)} \;\; u_{v,i+1} = u_{v,i} + \gamma(v_d - v_{i,e}) \end{aligned} \tag{4.7}$$

where $\gamma > 0$ is a learning gain, $u_{x,i}$ is the initial position and $u_{v,i}$ is the initial speed at the ith iteration.

Let u denote either initial speed or position, and z either final speed or position, from *Property* 4.1 we have

$$\begin{aligned} |u_d - u_{i+1}| &= |(u_d - u_i) - \gamma(z_d - z_i)| \\ &= ||u_d - u_i| - \gamma|z_d - z_i|| \,. \end{aligned} \tag{4.8}$$

To achieve learning convergence, a key issue is to determine the range of values for the learning gain γ, which is summarized in the following *Lemma*.

Lemma 4.1 *Suppose there exists a constant λ such that $|z_d - z_i| \leq \lambda|u_d - u_i|$, and there exists a $M < \infty$ such that $|u_d - u_1| = M$. For any given $\varepsilon > 0$, by applying the control law (4.7) and choosing the learning gain in the range*

$$\frac{1-\rho}{\lambda} < \gamma < \frac{1+\rho}{\lambda}, \qquad 0 < \rho < 1, \tag{4.9}$$

the output z_i will converge to the ε-neighbourhood of the desired output z_d with a finite number of iterations no more than

$$N = \frac{\log \dfrac{\varepsilon}{M\lambda}}{\log\left(1 - (1-\rho)\dfrac{\varepsilon}{M\lambda}\right)} + 1.$$

Proof: See *Appendix A.5*.

Remark 4.2 *The existence of a finite M can be easily verified as u_d is finite, and u_1 is always chosen to be a finite initial state in practical motion control problems.*

Remark 4.3 *The fastest convergence speed is $|1 - \gamma\lambda_i| = 0$ or $\gamma = 1/\lambda_i$. Since the specific value for λ_i is unknown to us, it would be more practical to discuss the relationship between the number of iterations N in Lemma 4.1 and the design parameters (ε, ρ).*

First let $\varepsilon \to 0$*, then*

$$\log \frac{\varepsilon}{M\lambda} \to -\infty$$

and

$$\log \left(1 - (1-\rho)\frac{\varepsilon}{M\lambda}\right) \to 0^{-}.$$

Consequently $N \to \infty$ *when* $\varepsilon \to 0$*, that is, the higher the precision, the larger the iteration number* N*.*

Next consider the design parameter ρ*. From (8.107) we can see that the convergence factor reaches minimum, or the learning speed reaches the maximum, when* $\rho \to 0$*. This property can also be derived from Lemma 4.1, because the magnitude of*

$$\log \left(1 - (1-\rho)\frac{\varepsilon}{M\lambda}\right)$$

is maximized when $\rho \to 0$*.*

In terms of *Lemma 4.1*, all we need to do is to find λ from the motion system so that the range of the learning gain γ can be determined. In Theorem 4.1, we derive the value of λ for all four cases.

Theorem 4.1 *The ILC convergence is guaranteed for cases (i) – (iv) when the learning gain is chosen to meet the condition (4.9), and the values of* λ *can be calculated respectively for four cases below.*

(i) In the initial position iterative learning for final position control, choose

$$\lambda = \exp\left(\int_{v_f}^{\mathcal{A}} L(v)dv\right).$$

(ii) In the initial speed iterative learning for final position control, choose

$$\lambda = \max_{v\in[v_f,\mathcal{A}]} g_1(v) \exp\left(\int_{v_f}^{\mathcal{A}} L(v)dv\right),$$

where g_1 is an upper bounding function satisfying $g(v,x) \leq g_1(v)$.

(iii) In the initial position iterative learning for final speed control, choose

$$\lambda = \frac{1}{c} \exp\left(\int_{v_d}^{\mathcal{A}} L(v)dv\right),$$

where c is a lower bound satisfying $0 < c \leq g(v,x)$.

(iv) In the initial speed iterative learning for final speed control, choose

$$\lambda = \frac{1}{c} \max_{v\in[v_d,\mathcal{A}]} g_1(v) \exp\left(\int_{v_d}^{\mathcal{A}} L(v)dv\right).$$

Proof: See *Appendix A.6*.

The prior knowledge required for four cases differs. The first case from position to position requires minimum prior knowledge from the motion system, the lower and upper bounds of $g(v,x)$ are not required. In the second case from speed to position, only the upper bounding function is required. In the third case from position to speed, only the lower bounding function is required. In the fourth case from speed to speed, however, both the lower and upper bounding functions are required.

Since g is the inverse of generalized damping or friction coefficient, the lower bound for g is to rule out the scenario where the generalized damping or friction coefficient would be infinity. Physically an overlarge damping or overlarge friction coefficient implies that an immediate stop-motion may occur, and we are unable to achieve the final speed control at a prespecified position x_f. Therefore the lower bound is required in cases (iii) and (iv) for final speed control.

The upper bound for g is required for initial speed learning to rule out the scenario where the generalized damping or friction coefficient would be too small. Look into the proof of case (ii), if the generalized damping or friction is too small, trajectories $\widehat{AB}$

and $\widehat{CD}$ will be very steep. As a result, a small change in the initial speed $\overline{CA}$ yields a significant position difference $\overline{AE}$. In other words, the system gain is extremely large and an extremely lower learning gain should be used. g_1 confines the system gain so that the lower bound of the learning gain can be determined.

4.4 A Dual Initial State Learning

In (4.1), consider such a scenario where f may drop to zero due to environmental changes, such as extremely low surface friction at certain places, meanwhile f could remain continuous with $v > 0$, $v_f > 0$ and $v_e > 0$. In such circumstances, it is appropriate to consider dv/dx in the phase plane

$$\frac{dv}{dx} = -\frac{f}{v} = -g(x, v), \tag{4.10}$$

where the generalized damping or friction coefficient is $g(x, v) = f/v$. Comparing with (4.2), in the dual problem (4.13) the positions of x and v are swopped, x is the argument and v is a function of x and the control inputs. The control tasks remain the same as the final position or speed control by means of the initial position or speed tuning. Thus the analysis in Theorem 4.1 can be directly extended to this dual scenario because *Assumption* 4.1 does not change and *Assumption* 5.1 holds with x and v swapped. Since the two control problems associated with (4.2) and (4.13) are the same except for the swapping between x and v, by employing the same ILC algorithms (4.7), the learning convergence properties for the four cases can be derived in a dual manner by swapping x_i with v_i, x_d with v_d, x_f with v_f, as summarized in Theorem 4.2.

Theorem 4.2 *The ILC convergence is guaranteed for cases (i) – (iv) when the learning gain is chosen to meet the condition (4.9), where the value of λ can be calculated respectively for four cases.*

(i) In initial position iterative learning for final position control, choose

$$\lambda = \frac{1}{c} \max_{x\in[0,x_d]} g_1(x) \exp\left(\int_0^{x_d} L(x)dx\right).$$

(ii) In initial speed iterative learning for final position control, choose

$$\lambda = \frac{1}{c} \exp\left(\int_0^{x_d} L(x)dx\right).$$

(iii) In initial position iterative learning for final speed control, choose

$$\lambda = \max_{x\in[0,x_f]} g_1(x) \exp\left(\int_0^{x_f} L(x)dx\right).$$

(iv) In initial speed iterative learning for final speed control, choose

$$\lambda = \exp\left(\int_0^{x_f} L(x)dx\right).$$

4.5 Further Discussion

Initial state learning in final state control implies that motion time and motion consumption for moving from initial desired state to final desired state are definite. As a compensation, it is meaningful to further consider the optimization among these indices. For example, try to shorten motion time but not increase motion consumption too much. Such optimization can be realized based on initial state learning and the following feedback learning.

4.5.1 Feedback learning control

Consider a class of motion systems with manipulated variables

$$-kh(x,v), \tag{4.11}$$

which denote the feedback information that is related to system states, where $h(x,v) \geq 0$ and $h = 0$ iff $v = 0$. In such circumstance, the control input k can be set as the constant feedback gain. Different from learning the initial state in final state control, an alternative approach with ILC is proposed to achieve final state control through adjusting the feedback coefficient iteratively. In this case, the dynamic system (4.1) becomes

$$\begin{cases} \frac{dx}{dt} &= v, \\ \frac{dv}{dt} &= -f(x,v) - kh(x,v). \end{cases} \tag{4.12}$$

Note that $v = 0$ are still equilibria of system (4.12). Without loss of generality, consider final speed control only later. Similarly as before, instead of considering system (4.12) directly, we may consider the system

$$\frac{dv}{dx} = -\frac{f}{v} - k\frac{h}{v} := -g(x,v,k),\ v(0,k) = \mathcal{A} \tag{4.13}$$

under *Assumption* 4.1 and the following *Assumption* 4.3.

Assumption 4.3 *For* $x,\ v_1,\ v_2 \in \mathbf{R}_+$, *and* $k_1,\ k_2 \in \mathbf{R}$, *there exist two known integrable bounding functions* $L_1(x)$ *and* $L_2(x)$ *such that*

$$|g(x,v_1,k_1) - g(x,v_2,k_2)| \leq L_1(x)|v_1 - v_2| + L_2(x)|k_1 - k_2|.$$

Define a final speed control

$$k_{i+1} = k_i - \gamma\left(v_d - v_{i,e}\right), \qquad k_0 = 0, \tag{4.14}$$

where $\frac{1-\rho}{\lambda} < \gamma < \frac{1+\rho}{\lambda},\ 0 < \rho < 1,\ \lambda = \exp\left(\int_0^{x_f} L_1 dx\right) \int_0^{x_f} L_2 dx$.

Theorem 4.3 *Assume* $v_d > 0$. *For system (4.13), under Assumptions 4.1 and 4.3, the learning control law (4.14) implies a desired control input* k_d *satisfying*

$$v(x_f, k_d) = v_d. \tag{4.15}$$

Proof: See *Appendix A.7.*

4.5.2 Combined initial state learning and feedback learning for optimality

Now, consider system

$$\begin{cases} \frac{dx}{dt} = v, & x(0) = 0, \\ \frac{dv}{dt} = -f(x,v) - kh(x,v), & v(0) = u. \end{cases} \tag{4.16}$$

Final state control can be performed by tuning the initial state u and feedback variable gain k together. Because the dimensions of the solution space are larger than that of the task space, the solution (u, k) is not unique. The extra degree of freedom offers an opportunity to find the "best" solution that meets certain optimization criteria. Analogous to optimization or optimal control problems, an objective function is introduced for final state control. In this work we consider the objective function

$$J(u,k) = \mathcal{L}_1(e) + \mathcal{L}_2(u,k) + \mathcal{L}_3(u,k) \tag{4.17}$$

where $\mathcal{L}_1(e)$ is a penalty to the final speed error, $\mathcal{L}_2(u,k)$ is a penalty to the energy consumption in the whole process and $\mathcal{L}_3(u,k)$ is corresponding to motion time.

A widely adopted penalty for the speed error is a quadratic function $\mathcal{L}_1(e) = c_1[v_d - v_{i,e}]^2$ with the weighting factor $c_1 > 0$. In the motion process, the energy consumption $\mathcal{L}_2$ is directly proportional to the integration of the square of speed v along the displacement x, thus the penalty $\mathcal{L}_2(u,k)$ can be chosen to be

$$\frac{c_2}{2}\int_0^{x_d} (v(u,k))^2\, dx$$

with the weighting factor $c_2 > 0$. Besides to minimize the energy consumption for the whole process, motion time is another important index we need to minimize. Thus,

it is feasible to let $\mathcal{L}_3(u,k) = c_3T$ with T being the motion time consumed from the initial state to final state and $c_3 > 0$. It is worthy to point out that equipment such as chronograph and speed visualizer is necessary for obtaining the objective function value for each u and k since the uncertainty of system induces that such value cannot be estimated only from the final state of motion systems.

The optimization problem (4.17) can be solved either analytically or numerically. However, both analytical and numerical methods require the system gradient components $D_1 = \partial J/\partial u$ and $D_2 = \partial J/\partial k$, thus require the exact model knowledge and other more information. Hence it is difficult to directly solve the optimization problem (4.17) arising from the combined final state control. Note that in preceding motion control, iterative learning can find appropriate control inputs without knowing the exact model and needs only the bounding knowledge of the system gradient. It would be interesting to explore iterative learning approach to solve the optimization problem (4.17), when only the bounding information of the gradient is available. Let α_j and β_j denote the lower and upper bounds of D_j, $j = 1, 2$ for any u or k. Choose the iterative learning control law

$$\begin{aligned} u_{i+1} &= u_i - \gamma_{1,i} J_i \\ k_{i+1} &= k_i - \gamma_{2,i} J_i, \end{aligned}$$

where $J_i \triangleq J(u_i, k_i)$. By substitution of the learning law

$$\begin{aligned} J_{i+1} &= J_i + (J_{i+1} - J_i) \\ &= J_i + D_{1,i}(u_{i+1} - u_i) + D_{2,i}(k_{i+1} - k_i) \\ &= (1 - \gamma_{1,i} D_{1,i} - \gamma_{2,i} D_{2,i}) J_i. \end{aligned}$$

Since $J_i > 0$, a minimum J can be reached if $J_{i+1} < J_i$. The key step in the optimal problem solving is to properly choose the learning gains $\gamma_{1,i}$ and $\gamma_{2,i}$ such that $0 \leq 1 - \gamma_{1,i} D_{1,i} - \gamma_{2,i} D_{2,i} < 1$. Let us consider two scenarios associated with the gradient

that is iteration dependent.

First consider the scenario where the gradient does not change signs for all u_i and k_i. Therefore the property $\alpha_j \leq D_{j,i} \leq \beta_j < 0$ or $0 < \alpha_j \leq D_{j,i} \leq \beta_j$, $j = 1, 2$, holds for all iterations. The learning gains can be designed to satisfy the convergence condition. For illustration, assume $\alpha_1 \leq D_{1,i} \leq \beta_1 < 0$ and $0 < \alpha_2 \leq D_{2,i} \leq \beta_2$. We can choose $\gamma_{1,i} = 1/2\alpha_1$ and $\gamma_{2,i} = 1/2\beta_2$. In the sequel

$$0 \leq 1 - \frac{D_{1,i}}{2\alpha_1} - \frac{D_{2,i}}{2\beta_2} < 1$$

is guaranteed. Note that the above quantity equals 1 only when $D_{1,i} = D_{2,i} = 0$, namely a minimum point of J_i is reached. When the signs of the gradient components are unknown, more learning trials can be conducted to detect the signs of the gradient components. Choose the learning gains

$$\gamma_{j,i} = \pm \frac{1}{2 \max\{|\alpha_j|, |\beta_j|\}} \quad j = 1, 2.$$

The pair $(\gamma_{1,i}, \gamma_{2,i})$ can take 4 sets of signs $\{1, 1\}$, $\{-1, 1\}$, $\{-1, -1\}$ and $\{1, -1\}$, corresponding to all possibles signs of the gradient $(D_{1,i}, D_{2,i})$. Therefore, at least one pair of learning gains satisfy

$$0 \leq 1 - D_{1,i}\gamma_{1,i} - D_{2,i}\gamma_{2,i} < 1$$

and $J_{i+1} < J_i$. Although the signs of $(D_{1,i}, D_{2,i})$ may not be available, by at most 4 trials it is guaranteed to find a set of $(\gamma_{1,i}, \gamma_{2,i})$ with consistent signs. Note that we may encounter situations with one correct gain and one wrong gain however still lead to a convergent result when the correct action overwhelms the wrong action. However this situation cannot guarantee the convergence for all subsequent learning iterations, because the magnitudes of gradient components may vary with respect to iterations.

Therefore it is necessary to consider all four gain sets to find the one with the greatest descending direction.

Next consider the scenario where the gradient components may change signs at different iterations. In such circumstances, the learning gains, once learned, may not be correct for subsequent iterations. Thus more learning trials are needed to detect the signs of the gradient components. Similar discussion can be done as in the above cases.

4.6 Illustrative example

Consider system (4.4) with parameters $m = 1,\ f_c = 3.5,\ f_s = 3.65,\ f_v = 1.06,\ v_s = 0.1,\ \delta = 0.05$. The target is to bring the motion system to a final state $(x_d, v_f) = (20, 0)$, i.e., let the motion system reach a displacement 20 m and stop. Since g is independent of x, Lipschitz function $L(v)$ is chosen to be zero.

Note that

$$g(v,x) = \frac{mv}{(f_c + (f_s - f_c)e^{-(\frac{v}{v_s})^{\delta}} + f_v v} < \frac{m}{f_v},$$

holds for any values of v, we can choose the upper bounding function $g_1(v) = \frac{m}{f_v} = 0.9434$. In terms of Theorem 4.1, when applying initial position learning which is case (i), $\lambda = 1$; and when applying initial speed learning which is case (ii), $\lambda = 0.9434$. The ILC law is given by (i) or (ii) in (4.7). In this example, choose the factor $\rho = 0.4$. According to Theorem 4.1, $0.6 < \gamma < 1.4$ for (i) initial position learning and $0.64 < \gamma < 1.48$ for (ii) initial speed learning. The ε-neighbourhood is chosen with $\varepsilon = 0.001$ m. Now, set a uniform learning gain $\gamma = 0.95$ and the learning results are shown in Fig.4.1 and Fig.4.2. In both cases, a quick learning convergence is achieved after repeating the learning process a few iterations.

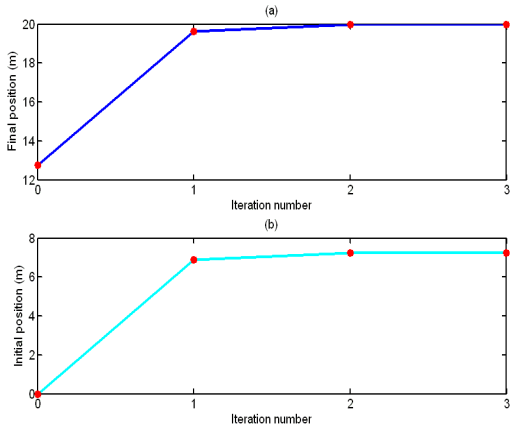


Figure 4.1: Initial position learning for final position control: $u_{x,1} = 0.0\ m$, $\mathcal{A} = 20.0$ m/s. (a) The observed final position; (b) The learning results of initial position.

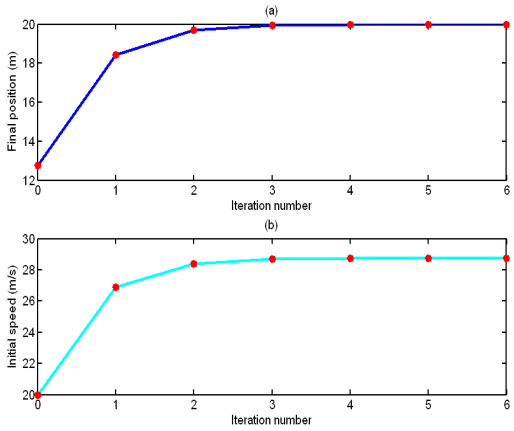


Figure 4.2: Initial speed learning for final position control: $u_{v,1} = 20.0$ m/s. (a) The observed final position; (b) The learning results of initial speed.

4.7 Conclusion

In this chapter we addressed a class of final state control problems for motion systems where the manipulated variables are initial states or feedback gains. Through iterative learning with the final state information, the desired initial states and/or feedback gains can be generated despite the existence of unknown nonlinear uncertainties in the motion systems. Both theoretical analysis and numerical simulations verify the effectiveness

of the proposed learning control schemes. It is worth pointing out that the learning convergence is achieved in terms of the absolute quantity or infinity norm, thus is a monotonic transient behavior in learning.

As another application of ILC, we consider the control problem of systems with input uncertainties in the next chapter, where the repetitiveness refers to the repetitiveness of control processes.

Chapter 5

A Dual-loop Iterative Learning Control for Nonlinear Systems with Hysteresis Input Uncertainty

5.1 Introduction

Recently, nonlinear system control with input uncertainties has received a great deal of attention, since input uncertainties are quite common phenomenon in engineering applications. Examples of input uncertainties include saturation, deadzone, hysteresis and so on. The existence of these input uncertainties may severely deteriorate the control performance or cause oscillations, even lead to system instability [118, 143–146].

ILC, as one of the well-known control techniques, has demonstrated its ability to deal with this sort of issue when the control environment is repeatable, see [32, 51, 87, 117, 143–146] for systems with saturation or deadzone nonlinearities. So far, however, much less work has been done to dynamic systems with hysteretic input uncertainty, although ILC design for hysteresis system has been frequently discussed [52, 53, 72, 79]. The difficulty in proving convergence of ILC algorithms for hysteretic systems arises due to two reasons: (i) branching effects and (ii) nonlinearity of each branch [21]. The latter

issue can be addressed by standard ILC methods. For example, the convergence of ILC on a single branch was shown in [52], in which the hysteresis nonlinearity was modeled as a single branch (using a polynomial). Alternatively, a functional approach was proposed for systems that satisfy the incrementally strictly increasing operator (ISIO) property [125]; however, the branching effect in hysteresis results in loss of the ISIO property [77]. The reason branching causes problems in proving convergence is because branching prevents the ILC algorithm from predicting the direction in which the input needs to be changed based on a measured output error. In [71], this problem has been addressed by constructing the monotonic property between input and output for a Preisach model.

Hysteresis is a very complex phenomenon and there exist many hysteresis models in literature, e.g., the Bouc-Wen model, Duhem model, the Jiles-Atherton model, the Prandtl-Ishlinskii model, and the Preisach model. A fact is that almost all the previous ILC design schemes are focused on the Preisach model, if hysteresis is addressed. However, as another typical class, the Bouc-Wen model for smooth hysteresis has received an increasing interest due to its capability to capture in an analytical form a range of shapes of hysteretic cycles which match the behavior of a wide class of hysteretic systems [56–58, 92, 105, 112]. In particular, it has been used experimentally to model piezoelectric elements, magnetorheological dampers, wood joints and base isolation devices for buildings. The obtained models have been used either to predict the behavior of the physical hysteretic element or for control purposes.

In this chapter, we address the ILC problem for a nonlinear dynamic system with a hysteresis input uncertainty, which takes the structure of the Bouc-Wen model. In normal cases, this class of systems does not show any standard cascaded structure. Thus, the method of backstepping design will lose its efficiency here, especially when consider-

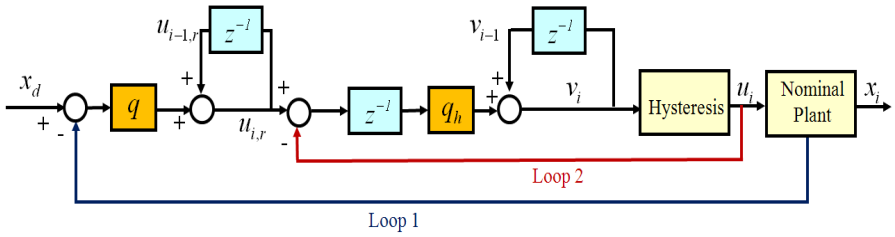


Figure 5.1: The schematic block diagram of the dual ILC loop. The operator z^{-1} denotes one iteration delay, and q, q_h are the learning gains for two sub-loops respectively.

ing convergence of tracking error along the iteration axis. Similarly as in [117], a dual iterative learning loop is applied to systems to learn both the unknown nominal dynamics and the input dynamics respectively, and then ensure the output of the system enter a prespecified neighborhood of the desired trajectory. More specifically, the first loop is a normal ILC scheme, which can guarantee the convergence of the output of dynamics without input uncertainty by using the composite energy function (CEF) method. Since this method is not a contraction mapping (CM) method, P-type ILC is proven to be enough for the system tracking; otherwise, D-type ILC should be adopted and the derivative information of tracking error in time domain must be available. The second loop is another ILC scheme to deal with the input dynamics. The input signal of loop 1 becomes the desired output for the loop 2. Subsequently, the ILC scheme in loop 2 drives the output of the input uncertainty to this desired output, achieved from loop 1. It should be noted that since both saturation and deadzone can be characterized as static mappings, many known iterative numerical algorithms are available for loop 2 to deal with input uncertainties. But for the hysteresis scenario, such an ILC law should be considered based on not a single mapping but its dynamics. After analyzing the input-output monotonicity property for the hysteretic input uncertainty, we can see that under certain conditions the simple ILC law is still efficient in this case. Based on the

convergence results for the two single loops separately, the convergence analysis for the dual-loop structure can then be discussed with the CEF method again.

When the strict input-output monotonicity does not hold for the hysteresis input part, the proposed ILC law can not work any more. This is because there exist a number of points on which the system gradient may vanish or change signs. In the sequel, we consider two more singular cases by adding a forgetting factor and incorporating a time-varying learning gain, and then ensure the corresponding ILC operator to be contractible. By using the Banach fixed-point theorem, we show that the output tracking error of the inner ILC loop (loop 2) can enter and remain ultimately in a sufficiently small neighborhood of zero. The dual-loop ILC convergence can then be discussed similarly as in the normal cases.

This chapter is organized as follows. Problem formulation is provided in Section 5.2. The ILC schemes for each single loop are presented in Sections 5.3 and 5.4, respectively. The convergence analysis for the dual-loop structure is considered in Section 5.5. Two singular scenarios are further considered in Section 5.6. To the end, illustrative examples are given in Section 5.7.

5.2 Problem Formulation

Consider the following SISO nonlinear dynamic system

$$\dot{x} = \eta(x,t) + u(v,z), \tag{5.1}$$

$$\dot{z} = D^{-1}(A\dot{v} - \beta|\dot{v}||z|^{n-1}z - \gamma\dot{v}|z|^n),\ z(0) = 0, \tag{5.2}$$

$$u = \alpha k v + (1-\alpha)Dkz, \tag{5.3}$$

where $v(t)$ and $x(t)$ are system input and output separately. The x-subsystem (5.1) with input u represents a nominal model, where $\eta(x,t) : \mathbf{R} \times \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}$ is a lumped

uncertainty, continuous in t and global Lipschitz continuous in x, i.e.

$$|\eta(x_1,t)-\eta(x_2,t)| \leq L_\eta |x_1-x_2|,$$

where L_η is a Lipschitz constant. The z-subsystem (5.2)-(5.3) denotes a physical model with a hysteretic component that takes the so-called Bouc-Wen type, where the unknown parameter vector $\boldsymbol{\theta} = \{n, A, D, k, \alpha, \beta, \gamma\}$ satisfying [56]

$$n \geq 1,\ A \neq 0,\ D > 0,\ k > 0, 0 \leq \alpha < 1,\ \beta + \gamma \neq 0. \tag{5.4}$$

Among these model parameters, D is the yield constant displacement and α is the postto pre-yielding stiffness ratio. The hysteretic part involves a nondimensional auxiliary variable z, which is the solution of the nonlinear first-order differential Eq. (5.2). In this equation, A, β, γ and n are nondimensional parameters, which control the shape and the size of the hysteresis loop [59]. The control task for system (5.1)-(5.3) is to drive the output signal $x(t)$ to track the reference signal $x_r(t), t \in [0,T], T > 0$, which is uniquely determined by the following dynamics

$$\dot{x}_r = \eta(x_r,t) + u_r, \tag{5.5}$$

$$\dot{z}_r = D^{-1}(A\dot{v}_r - \beta|\dot{v}_r||z_r|^{n-1}z_r - \gamma\dot{v}_r|z_r|^n), \tag{5.6}$$

$$u_r = \alpha k v_r + (1-\alpha)Dkz_r,\ z_r(0) = 0. \tag{5.7}$$

Here, we assume the system state x and the output of hysteresis dynamics u are measurable, and the control task would repeat itself along iteration axis in $[0,T]$. This kind of control issue is frequently encountered in many industrial processes, such as assembly lines and chemical batch processes. Intuitively, the information achieved from last iteration would be useful to improve the control performance in current iteration. However, due to the highly nonlinearity and severe uncertainty in system, not only in the nominal

part but also in the hysteresis input part, it is challenging to design an ILC law to realize the tracking task in a simple way.

In the first step, the plant (5.1)-(5.3) is decomposed into two sub-control systems:

$$\dot{x} = \eta(x,t) + u(t), \tag{5.8}$$

and

$$\dot{z}(t) = D^{-1}(A\dot{v} - \beta|\dot{v}||z|^{n-1}z - \gamma\dot{v}|z|^n), \quad z(0) = 0, \tag{5.9}$$

$$u(t) = \alpha k v + (1-\alpha)Dkz. \tag{5.10}$$

If regarding $u(t)$ as the input of system (5.8), an ILC control law can be designed directly. Thus, the second control system (5.9)-(5.10) should be ignored during the nominal control design for (5.8) because it destroys the feedback structure of the x-subsystem. On the other hand, different from saturation or deadzone case, the subsystem (5.9)-(5.10) does not take a static form but a dynamic one when denoting $v(t)$ and $u(t)$ as input and output respectively. Even this, we may prove in the following that a simple ILC law is still efficient to achieve tracking task for output $u(t)$. In the sequel, the dual-loop ILC for system (5.1)-(5.3) can be designed to achieve the real tracking for output $x(t)$.

5.3 Iterative Learning Control for Loop 1

The aim of loop 1 ILC design is to find a sequence of $u_i(t)$ to ensure the perfect tracking performance for (5.1) only. In other words, when there is no input uncertain dynamics, namely $u = u(v,z) = v$, the ILC scheme in loop 1 updates $u_i(t)$ iteratively to achieve perfect tracking performance. To facilitate the ILC design for (5.1), the following identical initialization condition (i.i.c.) is assumed.

Assumption 5.1 *$x_i(0) = x_0(0) = x_r(0)$, for all $i \in \mathbf{N}$.*

Remark 5.1 *From the practical point of view, the i.i.c. condition may not be satisfied. A possible way to solve the problem is to modify the target trajectory at the initial stage by making an appropriate interpolation [115], in the sequel guarantee* $x_r(0) - x_i(0) = 0$.

Denote the tracking error at i-th iteration as $e_i(t) = x_r(t) - x_i(t)$, the error dynamics of the system should be

$$\dot{e}_i = (\eta(x_r, t) + u_r) - (\eta(x_i, t) + u_i), \tag{5.11}$$

where $u_r \triangleq u(v_r, z_r)$ is the desired input in this loop.

Usually, there are two types of proof methods for ILC input or output error convergence, one is based on contraction mapping (CM) and the other one is based on Lyapunov functional, e.g., the composite energy function (CEF). If the relative degree is zero, generally, only the CM method is suitable, and monotonic convergence with the time weighted norm along the iteration axis can then be derived. If the relative degree is one, both the CM method and the CEF method can be used to derive the error convergence. The difference between them lies in that: the CM method requires the uniqueness of desired input signal, and a D-type ILC should be adopted over there so as to achieve a pointwise convergence; the CEF method does not need the uniqueness of desired input signal but its existence, and asymptotical convergence along the iteration axis can be guaranteed.

In order to avoid using the derivative information of output tracking error signal, which may not be accessible in real application, a simple ILC scheme is used in loop 1 as follows

$$u_i(t) = u_{i-1}(t) + qe_i(t), \quad t \in [0, T], \tag{5.12}$$

where $q > 0$ is the learning gain. The following theorem shows that the perfect tracking can be achieved by using (5.12).

Theorem 5.1 *Assume that Assumption 5.1 holds true for system (5.1). Then, the perfect tracking performance is achieved with the updating law (5.12).*

Proof: See *Appendix A.8*.

The updating law (5.12) iteratively modifies $u_i(t)$, which is actually the output of the hysteretic system (5.2)-(5.3), to realize tracking in loop 1. However, owing to the uncertainties in input dynamics, it is almost impossible to choose an appropriate $v_i(t)$ such that $u_i = u(v_i(t), z_i(t))$, without any additional effort. In the next part, another ILC loop is presented to address this point.

5.4 Iterative Learning Control for Loop 2

Consider the ILC for the hysteretic subsystem (5.9)-(5.10) under parametric condition (5.4), which can be regarded as the second loop for the whole system (5.1)-(5.3). In this loop, the desired input and output should be v_i and u_i respectively, where the latter is actually the input of loop 1 in the i-th iteration given by the ILC law (5.12). In the sequel, we use $v_{i,r}$ and $u_{i,r}$ to denote them to avoid confusion.

5.4.1 Preliminaries

A true physical hysteretic element should be BIBO stable, which means that for any bounded input signal $v(t)$ the hysteretic response is also bounded. Thus, the Bouc-Wen model (5.10) should keep the BIBO stability property in order to be considered as an adequate candidate to model real physical systems. Moreover, the boundedness and smoothness of internal variable $z(t)$ should also be addressed. Reference [56] concerned

these points and gave the following result.

Lemma 5.1 *Let $v(t), t \in [0, T]$ be a C^1 input signal. Assume the parametric condition (5.4) holds. Then, the Bouc-Wen model (5.9)-(5.10) is BIBO stable if one of the following four scenarios holds.*

$$(C_1): \quad A > 0, \quad \beta + \gamma > 0, \quad \textit{and} \quad \beta - \gamma \geq 0,$$

$$(C_2): \quad A > 0, \quad \beta - \gamma < 0, \quad \textit{and} \quad \beta \geq 0,$$

$$(C_3): \quad A < 0, \quad \beta - \gamma > 0, \quad \textit{and} \quad \beta + \gamma > 0,$$

$$(C_4): \quad A < 0, \quad \beta + \gamma < 0, \quad \textit{and} \quad \beta \geq 0.$$

For all the classes $(C_1) - (C_4)$, the signal $z(t)$ is C^1 bounded. Additionally, $|z(t)|$ is upper bounded by $z_a \triangleq \sqrt[n]{\frac{A}{\beta+\gamma}}$ in classes (C_1) and (C_2), while $|z(t)|$ is upper bounded by $z_b \triangleq \sqrt[n]{\frac{A}{\gamma-\beta}}$ in classes (C_3) and (C_4).

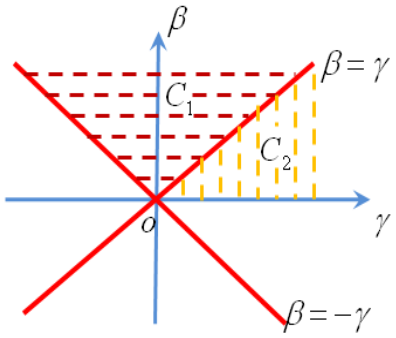


Figure 5.2: Graphic illustration of conditions C_1 and C_2 in γ-β plane, as $A > 0$.

Figs. 5.2 and 5.3 give the graphic illustration of the condition sets C_1-C_4 in γ-β plane. Actually, the parametric condition (5.4) represents all the normal cases and two limit cases ($n = 1$ and $\alpha = 0$) for the Bouc-Wen model. Noticing the i.i.c. for z, $z(0) = 0$, the BIBO property of hysteresis and the boundedness of $z(t)$ for the normal cases can be achieved directly from [56, *Theorem* 2] while the result corresponding to $n = 1$ and $\alpha = 0$

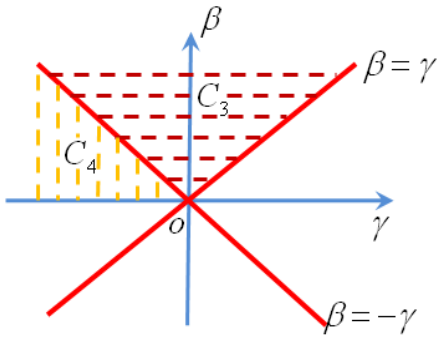


Figure 5.3: Graphic illustration of conditions C_3 and C_4 in γ-β plane, as $A < 0$.

was discussed in the Limit Cases analysis therein. The C^1 property of $z(t)$ can be derived similarly as discussed in [56, *Theorem* 3]. In detail, a state-space system realization of (5.9) is

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{z} &= D^{-1}(Ax_2 - \beta|x_2||z|^{n-1}z - \gamma x_2|z|^n), \end{aligned} \tag{5.13}$$

where $x_1 = v$. Since (5.13) is locally Lipschitz, a C^1 solution exists over some time interval $[0, t_0)$. We have seen that z and x_2 are bounded for every C^1 bounded signal v. This implies that the C^1 property of z can be extended to the interval $[0, T]$, only if $t_0 < T$.

More limit cases including $A = 0, \alpha = 1$, and $\beta + \gamma = 0$ are also analyzed briefly in [56], where the dynamics of hysteresis show some abnormal properties. For the case $A = 0$ or $\alpha = 1$, the hysteresis part in Eq. (5.9) is zero so that the system (5.10) is linear in v and thus does not give a hysteretic nonlinearity. For the case $\beta + \gamma = 0$, the upper bound on the variable z may depend on the input v, and thus the input-output gradient is a function of v. This will induce some difficulties when estimating the bound for gradient and subsequently choosing the learning gain for ILC law, if no prior bound information is available for v. Figs. 5.4-5.5 show the hysteretic behavior for such limit

case in phase plane and time domain respectively. In the next, we only consider the Bouc-Wen model under conditions (C_1)-(C_4).

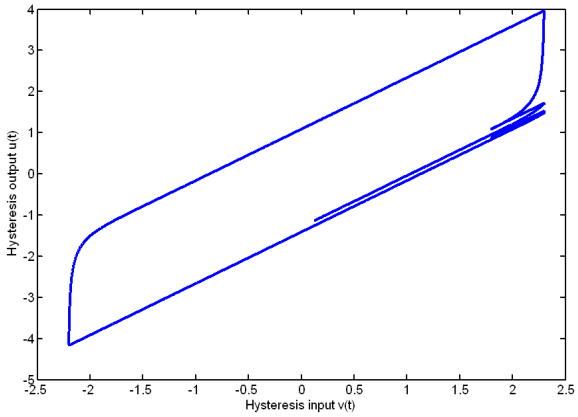


Figure 5.4: Hysteretic behavior with input $v(t) = 2\sin t + \cos 2t + 0.8, t \in [0, 10]$, for $\beta + \gamma = 0$. It can be seen that the input-output monotonicity still holds.

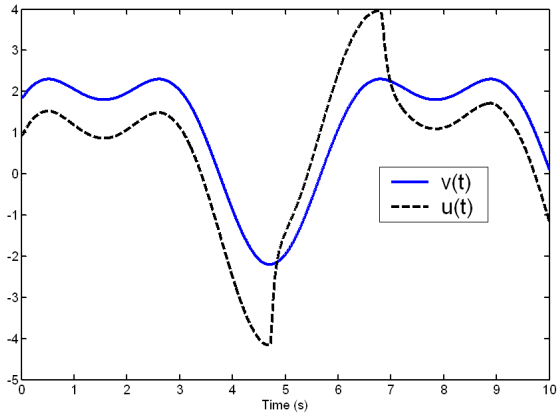


Figure 5.5: Profiles of input signal $v(t)$ and its corresponding output signal $u(t)$ in time domain for the hysteresis model as $\beta + \gamma = 0$.

Before proceeding to the convergence analysis of Loop 2, it is necessary to make clear how the input signal $v(t)$ affects the internal variable $z(t)$. Consider the equivalence of (5.9)

$$\dot{z} \quad = \quad \dot{v}\left[D^{-1}(A - \mathcal{S}(\dot{v})\beta|z|^{n-1}z - \gamma|z|^n)\right], \tag{5.14}$$

where $\mathcal{S}(\dot{v})$ takes the sign of $\dot{v}$, namely $\mathcal{S}(\dot{v}) = 1$ if $\dot{v} > 0$, $\mathcal{S}(\dot{v}) = -1$ if $\dot{v} < 0$, and $\mathcal{S}(\dot{v}) = 0$ otherwise. At each monotone branch of $v(t)$ satisfying $\mathcal{S}(\dot{v}(t)) = 1$, (5.14) is

$$\dot{z} = \cdot \left[D^{-1}(A - \beta|z|^{n-1}z - \gamma|z|^n) \right], \quad t \in (t_s, t_{s+1}), \tag{5.15}$$

where t_s and t_{s+1} are the nearest time instants backward and forward from time t satisfying $\dot{v}(t_s) = 0$ and $\dot{v}(t_{s+1}) = 0$. By multiplying dt on both sides of (5.15), it yields that

$$dz = dv \left[D^{-1}(A - \beta|z|^{n-1}z - \gamma|z|^n) \right],$$

or equivalently

$$\frac{dz}{dv} = D^{-1}(A - \beta|z|^{n-1}z - \gamma|z|^n). \tag{5.16}$$

Since the right hand side of (5.16) is continuous, locally Lipschitz, and $|z(t)|$ is always bounded by z_a or z_b, as stated in *Lemma* 5.1, there exists a unique solution for (5.16)

$$z(t) = f_1(v(t), v(t_s), z(t_s)), \quad t \in (t_s, t_{s+1}), \tag{5.17}$$

where $(v(t_s), z(t_s))$ is the initial condition for (5.16) at $t = t_s$. Because of the continuous dependence of solution $z(t)$ to initial values, the following equality uniformly holds for $v(t) \in (v(t_s), v(t_{s+1}))$,

$$\lim_{(\bar{v}(t_s), \bar{z}(t_s)) \to (v(t_s), z(t_s))} f_1(v(t), \bar{v}(t_s), \bar{z}(t_s)) = f_1(v(t), v(t_s), z(t_s)). \tag{5.18}$$

Similarly, at each monotone branch of $v(t)$ satisfying $\mathcal{S}(\dot{v}(t)) = -1, t \in (t_s, t_{s+1})$, (5.14) is

$$\dot{z} = \cdot \left[D^{-1}(A + \beta|z|^{n-1}z - \gamma|z|^n) \right], \tag{5.19}$$

whose unique solution is assumed to be

$$z(t) = f_2(v(t), v(t_s), z(t_s)), \quad t \in (t_s, t_{s+1}), \tag{5.20}$$

satisfying

$$\lim_{(\bar{v}(t_s),\bar{z}(t_s))\to(v(t_s),z(t_s))} f_2(v(t),\bar{v}(t_s),\bar{z}(t_s)) = f_2(v(t),v(t_s),z(t_s)) \tag{5.21}$$

uniformly for $v(t) \in (v(t_{s+1}), v(t_s))$.

Summarily, the solution of (5.14) corresponding to different monotone branches of input $v(t)$ can be written into a compact form

$$z(t) = f(v(t), v(t_s), z(t_s), \text{sign}(\dot{v}(t))), \quad t \in (t_s, t_{s+1}), \tag{5.22}$$

where

$$f = \begin{cases} f_1(v(t), v(t_s), z(t_s)), & \text{if } \mathcal{S}(\dot{v}(t)) = 1, \\ f_2(v(t), v(t_s), z(t_s)), & \text{if } \mathcal{S}(\dot{v}(t)) = -1. \end{cases} \tag{5.23}$$

According to *Lemma* 5.1, under the parametric conditions (C_1)-(C_4), $z(t)$ is C^1 bounded if $v(t)$ is C^1. Noticing the hysteresis output expression (5.10), $u(t)$ is also a function of $v(t), v(t_s), z(t_s)$, and $\mathcal{S}(\dot{v}(t))$ at each monotonic branch of input $v(t)$, i.e.,

$$u(t) = u(v(t), v(t_s), z(t_s), \mathcal{S}(\dot{v}(t))), \quad t \in (t_s, t_{s+1}). \tag{5.24}$$

5.4.2 Input-output gradient evaluation

Define the ILC law as follows

$$v_j(t) = v_{j-1}(t) + q_h \Delta u_{j-1}(t), \quad t \in [0, T], \tag{5.25}$$

where $\Delta u_{j-1} = u_{i,r} - u_{j-1}$, and $q_h > 0$ is the constant ILC gain. It is known that to determine the gain q_h is highly related to the system information, or input-output gradient information. Therefore, in order to achieve the learning convergence of (7.4), it is necessary to estimate in advance the gradient of u with respect to v in the presence of system uncertainties. In general, most of the ILC applications are based on input-output

monotonicity, i.e., a monotonic input (in time) causes a monotonic output (in time). This means that the sign of input-output gradient is always positive or negative. For a preisach type of hysteresis, the input-output monotonicity is assumed in [71, 72], and the bound information of gradient is given by the following inequalities

$$d_1(v_2 - v_1)^n \leq (u_2 - u_1) \leq d_2(v_2 - v_1), \tag{5.26}$$

where $d_1, d_2 > 0$ are constants, and n is a positive integer. Here, we present the corresponding result on the Bouc-Wen model.

Lemma 5.2 *Let*

$$\begin{aligned} (C_3'): \quad & (C_3) \quad \textit{and} \quad \frac{\alpha}{1-\alpha} + \frac{2\beta A}{\beta - \gamma} \geq \frac{\epsilon}{(1-\alpha)k}, \\ (C_4'): \quad & (C_4) \quad \textit{and} \quad \frac{\alpha}{1-\alpha} + A \geq \frac{\epsilon}{(1-\alpha)k}, \end{aligned} \tag{5.27}$$

where $0 < \epsilon \leq \alpha k$ is an arbitrarily small constant. The graphic illustration of these two sets of conditions in γ-β plane are shown in Fig. 5.6. Under one of the cases $(C_1), (C_2), (C_3')$ and (C_4'), the Bouc-Wen model (5.9)-(5.10) possesses the input-output monotonicity in each monotone branch of $v(t)$, with uniform gradient bounds

$$0 < \epsilon \leq \frac{\partial u(t)}{\partial v(t)} \leq k \max\{1, \alpha, 2A\}. \tag{5.28}$$

Proof: See *Appendix A.9.*

5.4.3 Asymptotical learning convergence analysis

The difficulty in proving convergence of ILC algorithms for hysteretic systems arises due to two reasons: (i) branching effects and (ii) nonlinearity of each branch [21]. The latter issue can be addressed by standard ILC methods. However, the branching effect

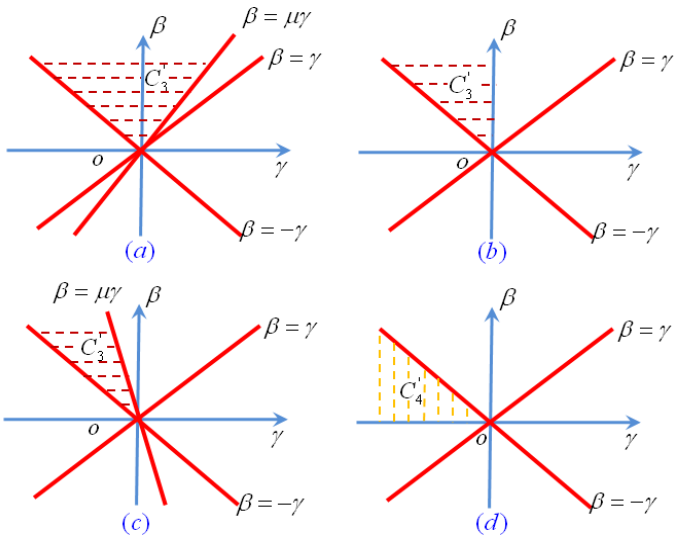


Figure 5.6: Graphic illustration of conditions C_3' and C_4' in γ-β plane, where μ satisfies $\mu\mu_1 = (\frac{\alpha}{1-\alpha} - \frac{\epsilon}{(1-\alpha)k}), \mu_1 = 2A + \frac{\alpha}{1-\alpha} - \frac{\epsilon}{(1-\alpha)k}$. (a): $\mu_1 > 0$. Then, the condition $\frac{\alpha}{1-\alpha} + \frac{2\beta A}{\beta-\gamma} \geq \frac{\epsilon}{(1-\alpha)k}$ is equivalent to $\beta \geq \mu\gamma (> \gamma)$. (b): $\mu_1 = 0$. The condition $\frac{\alpha}{1-\alpha} + \frac{2\beta A}{\beta-\gamma} \geq \frac{\epsilon}{(1-\alpha)k}$ is equivalent to $\gamma \leq 0$. (c): $\mu_1 < 0$ and $\mu \leq -1$. The condition $\frac{\alpha}{1-\alpha} + \frac{2\beta A}{\beta-\gamma} \geq \frac{\epsilon}{(1-\alpha)k}$ is equivalent to $\beta \leq \mu\gamma (\leq -\gamma)$. It is noted that C_3' is an empty set as $\mu_1 < 0$ and $\mu > -1$. (d): $0 > A \geq \frac{\epsilon}{(1-\alpha)k} - \frac{\alpha}{1-\alpha}$. Otherwise, the set C_4' is empty.

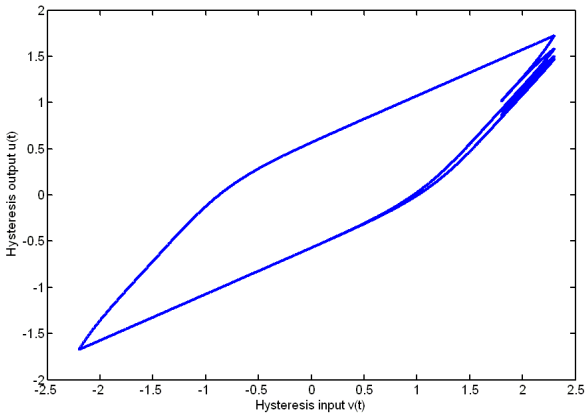


Figure 5.7: Class (C_1): Hysteretic behavior with input $v(t) = 2\sin t + \cos 2t + 0.8, t \in [0, 10]$ that satisfies the input-output monotonicity property, where $n = 3, A = 1.5, D = 1, k = 1, \alpha = 0.5, \beta = 0.9, \gamma = 0.1$, and the initial state is $(v(0), u(0)) = (1.8, 0.9)$.

in hysteresis would result in loss of the ISIO property [77]. The reason branching causes problems in proving convergence is because branching prevents the ILC algorithm from

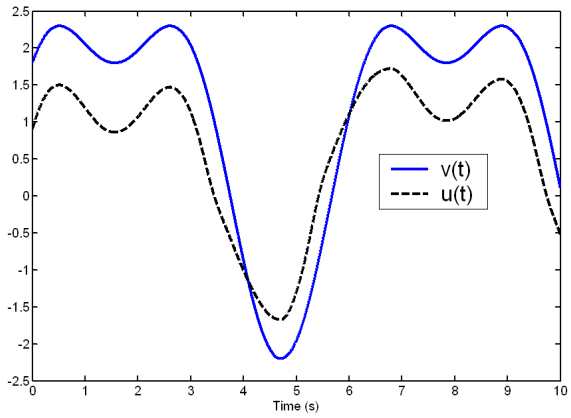


Figure 5.8: Class (C_1): Profiles of input signal $v(t)$ and its corresponding output signal $u(t)$ in time domain for the hysteresis model, where $n = 3, A = 1.5, D = 1, k = 1, \alpha = 0.5, \beta = 0.9$, $\gamma = 0.1$, and the initial state is $(v(0), u(0)) = (1.8, 0.9)$.

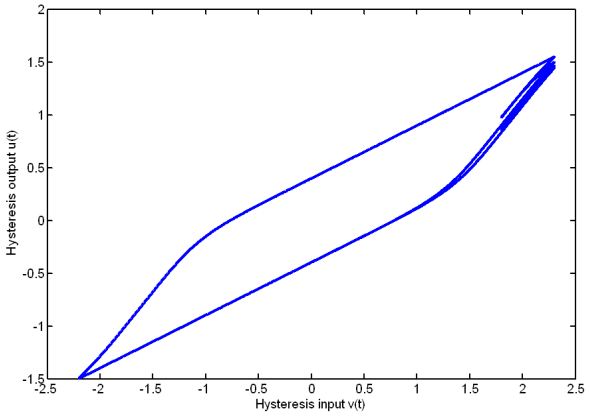


Figure 5.9: Class (C_2): Hysteretic behavior with input $v(t) = 2\sin t + \cos 2t + 0.8, t \in [0, 10]$ that satisfies the input-output monotonicity property, where $n = 3, A = 1.5, D = 1, k = 1, \alpha = 0.5, \beta = 0.9$, $\gamma = 2.1$, and the initial state is $(v(0), u(0)) = (1.8, 0.9)$.

predicting the direction in which the input needs to be changed based on a measured output error. Using the input-output monotonicity property in each branch, we can obtain the following lemma which can help us overcome the branching effect.

Lemma 5.3 *Consider the Bouc-Wen model (5.9)-(5.10) under either of the conditions from* (C_1), (C_2), (C_3') *to* (C_4'). *Given a desired* C^1 *bounded reference signal* $u_{i,r}(t), t \in$

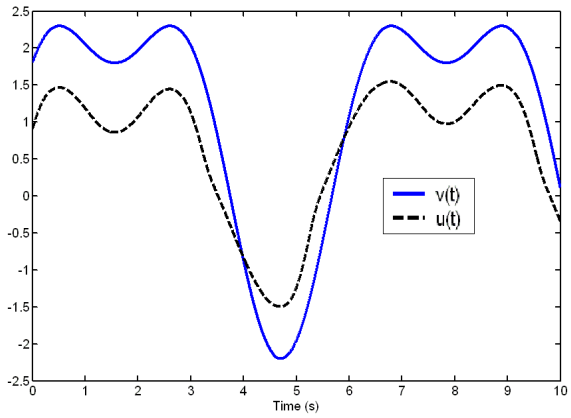


Figure 5.10: Class (C_2): Profiles of input signal $v(t)$ and its corresponding output signal $u(t)$ in time domain for the hysteresis model, where $n = 3, A = 1.5, D = 1, k = 1, \alpha = 0.5, \beta = 0.9$, $\gamma = 2.1$, and the initial state is $(v(0), u(0)) = (1.8, 0.9)$.

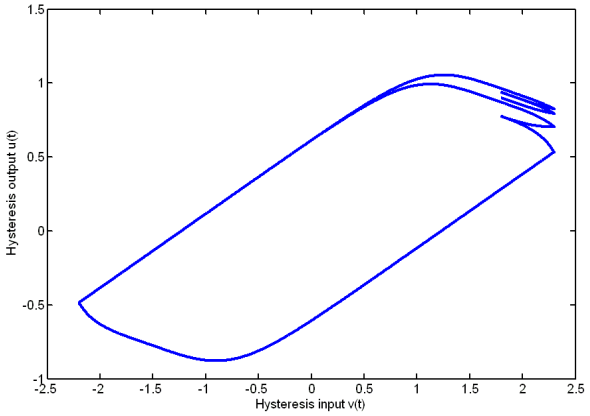


Figure 5.11: Class (C_3): Hysteretic behavior with input $v(t) = 2\sin t + \cos 2t + 0.8, t \in [0, 10]$ that does not satisfy the input-output monotonicity property, where $n = 3, A = -1.5, D = 1, k = 1, \alpha = 0.5, \beta = 0.9$, $\gamma = 0.1$, and the initial state is $(v(0), u(0)) = (1.8, 0.9)$.

$[0, T]$ that the hysteresis system is able to track. Let $\lambda = k \max\{1, \alpha, 2A\}$, and $0 < q_h \leq 1/\lambda$ for the ILC law (7.4). Assume the initial input $v_0(t), t \in [0, T]$ is C^1 bounded. If $\mathcal{S}(\dot{v}_0(t)) = \mathcal{S}(\dot{u}_{i,r}(t))$, then the four quantities u_j, v_j, $u_{i,r}$ and $v_{i,r}$ take same monotonic properties at time t for all $j = 0, 1, 2, \cdots$.

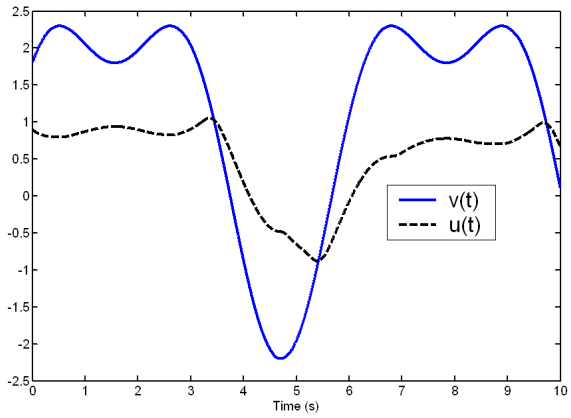


Figure 5.12: Class (C_3): Profiles of input signal $v(t)$ and its corresponding output signal $u(t)$ in time domain for the hysteresis model, where $n = 3, A = -1.5, D = 1, k = 1, \alpha = 0.5, \beta = 0.9$, $\gamma = 0.1$, and the initial state is $(v(0), u(0)) = (1.8, 0.9)$.

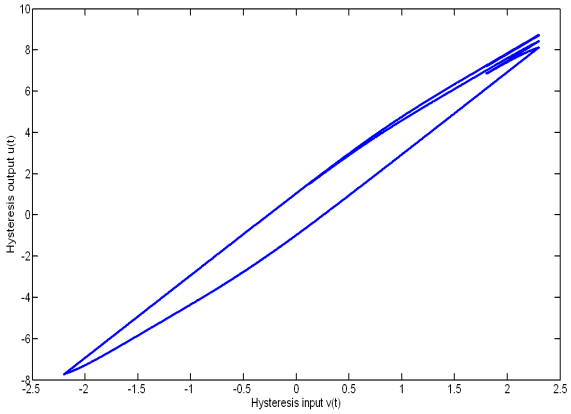


Figure 5.13: Class (C_3'): Hysteretic behavior with input $v(t) = 2\sin t + \cos 2t + 0.8, t \in [0, 10]$ that satisfies the input-output monotonicity property, where $n = 3, A = -1, D = 1, k = 5, \alpha = 0.8, \beta = 1.0$, $\gamma = 0.2$, and the initial state is $(v(0), u(0)) = (1.8, 7.2)$.

Proof: See *Appendix A.10*.

Based on the results in *Lemmas* 5.1, 5.2, and 5.3, we have

Theorem 5.2 *Consider the Bouc-Wen model (5.10) in one of the cases* (C_1), (C_2), (C_3'), *and* (C_4'). *If applying the control law*

$$v_j(t) = v_{j-1}(t) + q_h \Delta u_{j-1}(t),\ 0 < q_h \leq \frac{1}{\lambda},\ t \in [0, T], \tag{5.29}$$

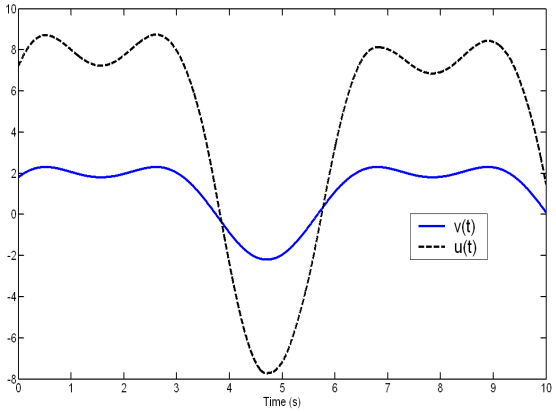


Figure 5.14: Class (C_3'): Profiles of input signal $v(t)$ and its corresponding output signal $u(t)$ in time domain for the hysteresis model, where $n = 3, A = -1, D = 1, k = 5, \alpha = 0.8, \beta = 1.0$, $\gamma = 0.2$, and the initial state is $(v(0), u(0)) = (1.8, 7.2)$.

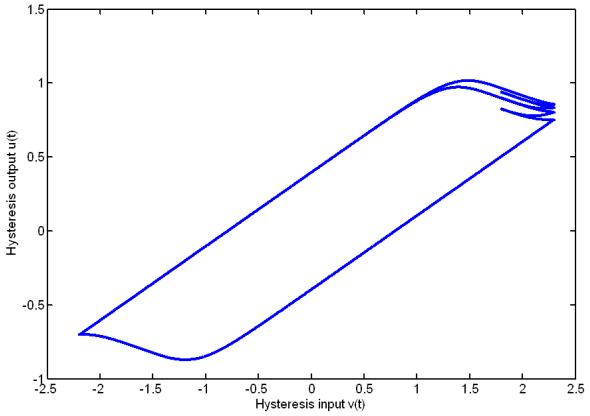


Figure 5.15: Class (C_4): Hysteretic behavior with input $v(t) = 2\sin t + \cos 2t + 0.8, t \in [0, 10]$ that does not satisfy the input-output monotonicity property, where $n = 3, A = -1.5, D = 1, k = 1, \alpha = 0.5, \beta = 0.9$, $\gamma = -2.1$, and the initial state is $(v(0), u(0)) = (1.8, 0.9)$.

where both the initial input $v_0(t)$ *and the desired output* $u_{i,r}(t)$ *are* C^1 *bounded and* $\lambda = k \max\{1, \alpha, 2A\}$*, then there exist* $\rho \in (0, 1)$ *such that*

$$|\Delta u_j| \leq \rho |\Delta u_{j-1}| + |\sigma_j(t)|, \tag{5.30}$$

where $\lim_{j\to\infty} \sigma_j(t) = 0$.

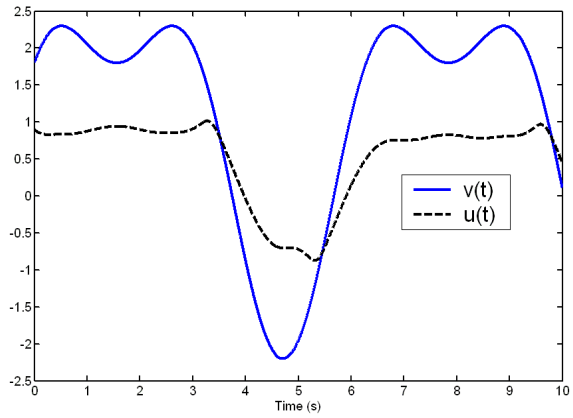


Figure 5.16: Class (C_4): Profiles of input signal $v(t)$ and its corresponding output signal $u(t)$ in time domain for the hysteresis model, where $n = 3, A = -1.5, D = 1, k = 1, \alpha = 0.5, \beta = 0.9$, $\gamma = -2.1$, and the initial state is $(v(0), u(0)) = (1.8, 0.9)$.

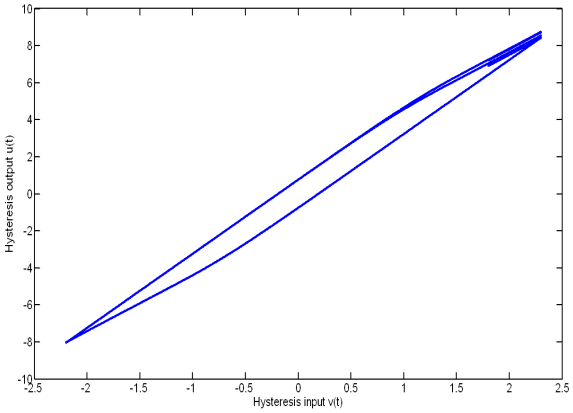


Figure 5.17: Class (C_4'): Hysteretic behavior with input $v(t) = 2\sin t + \cos 2t + 0.8, t \in [0, 10]$ that satisfies the input-output monotonicity property, where $n = 3, A = -1, D = 1, k = 5, \alpha = 0.8, \beta = 1.0$, $\gamma = -1.2$, and the initial state is $(v(0), u(0)) = (1.8, 7.2)$.

Proof: See *Appendix A.11*.

Based on the convergence results on the two single ILC loops, the CEF method is then adopted in the next section to derive the convergence of dual-loop ILC.

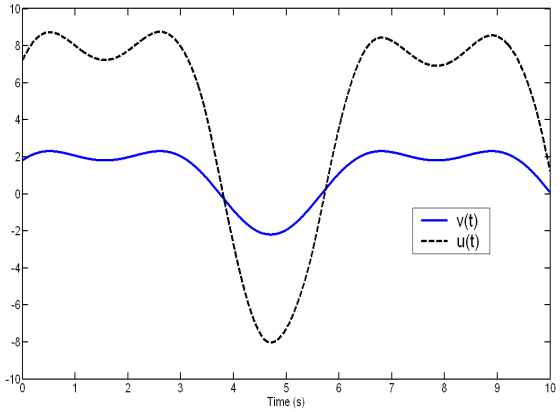


Figure 5.18: Class (C_4'): Profiles of input signal $v(t)$ and its corresponding output signal $u(t)$ in time domain for the hysteresis model, where $n = 3, A = -1, D = 1, k = 5, \alpha = 0.8, \beta = 1.0$, $\gamma = -1.2$, and the initial state is $(v(0), u(0)) = (1.8, 7.2)$.

5.5 Dual-loop Iterative Learning Control

In the preceding two sections, ILC laws have been presented for the x-subsystem and the hysteretic subsystem separately. Obviously, if these two ILC loops are only combined in a simple cascaded way, or in more detail if every input in the first loop is sought by infinite iterations in the second loop, this will make the control learning very inefficient, or even results in a failure. For saturation or deadzone input uncertainty, [117] combines these two simple loops into a dual loop: one is to learn the unknown nominal dynamics while the other is to learn the unknown actuator, and both of them are learned simultaneously. Specifically, the whole ILC law is constructed as follows.

$$u_{i,r}(t) = u_{i-1,r}(t) + qe_i(t), \tag{5.31}$$

$$\Delta u_i(t) = u_{i,r}(t) - u_i(t), \tag{5.32}$$

$$v_i(t) = v_{i-1}(t) + q_h \Delta u_{i-1}(t), \tag{5.33}$$

where $q > 0$, $0 < q_h \leq 1/\lambda$, $\lambda = k \max\{1, \alpha, 2A\}$, $u_{0,r} = 0$, $u_i(t) = u(v_i(t), z_i(t))$, $t \in [0, T]$. We will state that the dual ILC loop is still applicable for the hysteretic input

case, although the input uncertainty shows a dynamic behavior.

Theorem 5.3 *Let Assumption 5.1 and one of the sets* (C_1), (C_2), (C_3') *and* (C_4') *hold. For the system (5.1)-(5.3), using the dual ILC law (5.31-5.33) will yield that for any small* $\iota > 0$*, there exists a finite iteration number* i_ι *such that the output tracking error* $|e_i(t)| < \iota$ *for all* $i \geq i_\iota$ *and* $t \in [0, T]$.

Proof: See *Appendix A.12*.

Remark 5.2 *In this work, a simple scalar dynamic system is considered only. However, the dual-loop ILC structure can be further applied to more complex scenarios with slight modification, e.g., the input nonlinearity is a combination of saturation, deadzone, and saturation, or the system is multi-input-multi-output.*

5.6 Extension to Singular Cases

Up to now, we did not consider the scenarios in which (C_3) or (C_4) is satisfied but (C_3') and (C_4') are not. In these cases, the Bouc-Wen model may not keep the input-output monotonicity property, which however is essential to any ILC strategy. Moreover, the existence of positive parameter $\epsilon(\leq \alpha k)$ in *Lemma* 5.2 further confines our previous discussion into the strict monotonic case. In a strict point of view, when the strict input-output monotonicity does not exist in hysteresis, the system could become singular at a number of points and the learnability condition will be violated at those points [151]. In the following, two types of singularities are considered by considering the hysteretic part or the loop 2 part only. In the first situation, the system gradient does not change signs (the control direction) on the two sides of singular points or segments. For this case, we need merely to do a very minor modification to the typical ILC updating law by

adding a forgetting factor close to unity. Then, by the fixed point theorem, the revised contraction mapping will generate a control input sequence converging to a unique fixed point uniformly, and this fixed point warrants the hysteresis output to ultimately and uniformly enter a small neighborhood of the target trajectory. In the second situation, the gradient changes its sign on two sides of a singular point. Different from the first case, we have to get to know when a second type singularity occurs, and how the sign changes. For this case, in addition to the forgetting factor, we further incorporate the sign changes into the revised ILC operator. By using the revised ILC law, the control input sequence will converge uniformly to a unique fixed point, and the system enters a designed neighborhood of the target trajectory except for a number of subintervals centered about the second type singular points.

5.6.1 ILC for the first type of singularities

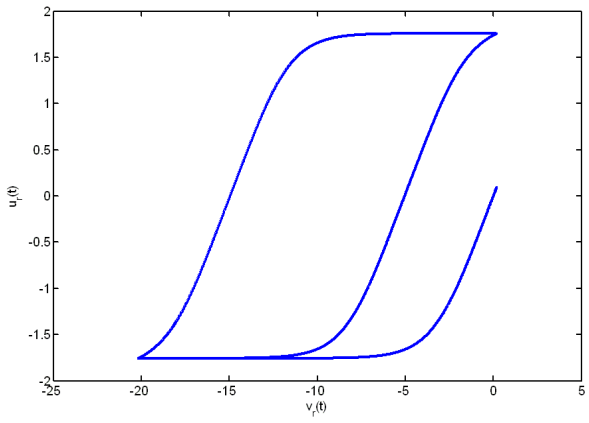


Figure 5.19: The first singular case: $\alpha = 0$, where the hysteresis behavior corresponds to the desired input $v_r(t) = \sin 2t + 10\cos t - 10, t \in [0, 10]$. It can be seen that $\dot{u}_r(t) = 0$ in certain intervals of $[0, T]$.

The Bouc-Wen model was originally developed in the context of mechanical systems in which v is a displacement and $u(v(t), z(t))$ is a restoring force. It represents the

hysteretic force u as the superposition of an elastic component αkv and a purely hysteretic component $(1-\alpha)kDz$. If we only take the pure hysteretic component into account or equivalently let $\alpha = 0$, the input-output monotonicity can be guaranteed as $A \neq 0$, or specifically $\partial u/\partial v \geq 0$ as $A > 0$ and $\partial u/\partial v \leq 0$ as $A < 0$ but its strictness will be violated in some segments. As a first extension, we consider this case in the following.

Now, only focus on the hysteretic part and the system (5.2)-(5.3) becomes

$$\begin{aligned} \dot{z} &= \dot{v}[D^{-1}(A - |z|^n(\gamma + \beta\mathcal{S}(\dot{v}z)))], \\ u &= kDz,\ z(0) = 0. \end{aligned} \tag{5.34}$$

Subsequently,

$$\dot{u} = \cdot \left(vkA - \frac{|u|^n}{k^{n-1}D^n}(\gamma + \beta\mathcal{S}(\dot{v}u))\right),\ u(0) = 0, \tag{5.35}$$

satisfying $0 \leq \partial u/\partial v \leq 2kA$ in (C_1) and (C_2), and $\min\{2kA\beta/(\beta-\gamma), kA\} \leq \partial u/\partial v \leq 0$ in (C_3) and (C_4) by similar discussion as in (8.44)-(8.49). The ILC algorithms corresponding to the four cases are

$$v_j(t) = (1-\zeta_0)v_{j-1}(t) + q_h\Delta u_{j-1}(t), \tag{5.36}$$

where ζ_0 is a constant satisfying $0 < \zeta_0 \ll 1$, $q_h > 0$ in (C_1) and (C_2) while $q_h < 0$ in (C_3) and (C_4). $\Delta u_{j-1} = u_r - u_{j-1}$, where u_r is uniquely determined by the desired input $v_r(t)$ and dynamics

$$\begin{aligned} \dot{u}_r &= \dot{v}_r\left(kA - \frac{|u_r|^n}{k^{n-1}D^n}(\gamma + \beta\mathcal{S}(\dot{v}_r u_r))\right), \\ u_r(0) &= 0. \end{aligned}$$

Different from the normal case, the hysteresis output $u(t)$ is only relative to the variation rate of the input, i.e. $\dot{v}(t)$, as shown in (5.35), although $v(t)$ is the real system input in

practice. Thus, it is rational to assume an i.i.c. in the ILC law (5.36), namely

$$v_j(0) = v_0(0) = \xi_v, j = 1, 2, \cdots. \tag{5.37}$$

Note that the value of constant ξ_v may be unknown to us and different from the initial value of desired input $v_r(0)$.

Theorem 5.4 *Consider the hysteresis dynamics (5.35) and let the magnitude of learning gain q_h satisfy*

$$0 < |q_h| \leq \frac{1 - \zeta_0}{\lambda} \tag{5.38}$$

where $\lambda = 2kA$ in (C_1) and (C_2) and $\lambda = -\min\{2kA\beta/(\beta - \gamma), kA\}$ in (C_3) and (C_4). Then, assuming v_0 is C^1 bounded, the ILC law (5.36) under condition (5.37) warrants a convergent sequence v_j to a unique fixed point v^*, and there exists a constant $0 < \rho < 1$ such that the output error Δu_j satisfies

$$|\Delta u_j(t)| \leq \rho |\Delta u_{j-1}(t)| + |\sigma_j(t)|, \quad t \in [0, T], \tag{5.39}$$

in which the function $\sigma_j(t)$ is bounded and

$$|\sigma_j(t)| \leq \frac{\zeta_0(1 - \rho)}{|q_h|} |v^*(t)|_s$$

as $j \to \infty$, where $|v^*(t)|_s$ represents the supreme norm of $v^*(t)$ over $[0, T]$.

Proof: See *Appendix A.13.*

Remark 5.3 *From (8.84), the residual part $\sigma_j(t)$ will approach to zero with an error of $\frac{\zeta_0(1-\rho)}{|q_h|}|v^*(t)|_s$, which could be small enough by tuning the parameters ζ_0 and ρ. By combining the results in Theorems 5.1 and 5.4, the convergence of dual loop ILC for the*

case of $\alpha = 0$ can be achieved similarly as in Theorem 6.1. The only difference is that in (8.76), i.e.,

$$\begin{aligned} E_i(t) &\leq E_0 + \left(\frac{1+\rho^2}{2q\left(1-\rho^2\right)\rho^2}\right)\int_0^t \sum_{p=1}^{i} |\sigma_p(\tau)|^2 d\tau \\ &\quad -\frac{1}{2}e^{-\zeta T}\sum_{p=0}^{i-1} e_{p-1}^2, \end{aligned} \tag{5.40}$$

σ_p does not vanish any more but converges to a small neighborhood of zero. As such, the positiveness of $E_i(t)$ implies that as $i \to \infty$

$$\frac{1+\rho^2}{2q\left(1-\rho^2\right)\rho^2}\int_0^t \frac{\zeta_0(1-\rho)}{|q_h|}|v^*(t)|_s d\tau \geq \frac{1}{2}e^{-\zeta T}e_i^2(t)$$

Accordingly, the dual-loop tracking error should be finally bounded by

$$\sqrt{Te^{\zeta T}\frac{1+\rho^2}{q\left(1+\rho\right)\rho^2}\frac{\zeta_0}{|q_h|}|v^*(t)|_s}.$$

Since $0 < \zeta_0 \ll 1$ is a tunable parameter, this bound could be very small.

5.6.2 ILC for the second type of singularities

As the parameter set (C_3) or (C_4) holds, the input-output monotonicity of the hysteretic subsystem may not hold, i.e., there exist singular points in which the following gradient changes signs.

$$\begin{aligned} &\frac{\partial u}{\partial v} = \alpha k + (1-\alpha)k \\ &\times\left(A - \left|\frac{u-\alpha k v}{(1-\alpha)Dk}\right|^n\left(\gamma + \beta\mathcal{S}\left(\frac{\dot{v}(u-\alpha k v)}{(1-\alpha)Dk}\right)\right)\right). \end{aligned} \tag{5.41}$$

Specifically, according to *Lemma* 5.2, in (C_3)

$$\alpha k + 2(1-\alpha)k\frac{\beta A}{\beta-\gamma} \leq \frac{\partial u}{\partial v} \leq k\alpha, \tag{5.42}$$

and in (C_4)

$$\alpha k + (1-\alpha)kA \leq \frac{\partial u}{\partial v} \leq k\alpha. \tag{5.43}$$

If $\alpha k + 2(1-\alpha)k\frac{\beta A}{\beta-\gamma} < 0$ in (C_3) or $\alpha k + (1-\alpha)kA < 0$ in (C_4), the second type of singularities occurs. For this scenario, more knowledge is needed for the desired gradient variation such that we can derive a smooth control gain function $q_h(t) \in \mathcal{C}^1([0,T],\mathcal{R})$ to ensure $q_h(t)\partial u_r/\partial v_r \geq 0$.

Theorem 5.5 *Consider the hysteresis dynamics (5.10) and the following ILC law*

$$v_j(t) \;=\; (1-\zeta_0)v_{j-1}(t) + q_h(t)\Delta u_{j-1}(t) \tag{5.44}$$

where $0 < \zeta_0 \ll 2/3$, $0 < |q_h(t)| \leq q_h^0 \leq \frac{\zeta_0}{2\lambda}, t \in [0,T]$, *and* λ *is the upper bound of the gradient, i.e.,*

$$\lambda = \max\left\{\left|\alpha k + 2(1-\alpha)k\frac{\beta A}{\beta-\gamma}\right|, k\alpha\right\}$$

in (C_3) *and*

$$\lambda = \max\left\{\left|\alpha k + (1-\alpha)kA\right|, k\alpha\right\}$$

in (C_4). *Divide the interval* $[0,T]$ *into two subsets:* $\Omega_1 = \left\{t \in [0,T] : |q_h(t)| = q_h^0\right\}$ *and* $\Omega_2 = [0,T] - \Omega_1$, *where* Ω_2 *is composed of a number of open sets, each covering a singular point* t_s *with its length* δ. *Then, assuming* v_0 *is* C^1 *bounded, the ILC law (5.44) warrants a convergent sequence* v_j *to a unique* C^1 *function* $v^*(t)$, *and there exists a constant* $0 \leq \rho < 1$ *such that the output error* Δu_j *satisfies*

$$|\Delta u_j(t)| \leq \rho|\Delta u_{j-1}(t)| + |\sigma_j(t)|, \quad t \in [0,T], \tag{5.45}$$

in which the function $\sigma_j(t)$ *is bounded and*

$$|\sigma_j(t)| \leq \frac{\zeta_0(1-\rho)}{|q_h^0|}|v^*(t)|_s + \delta(1-\rho)\sum_{i=1}^{2}\beta_i \tag{5.46}$$

as $j \to \infty$, *where the constants* β_i *satisfy*

$$|\dot{u}_r(t)| \leq \beta_1, \quad |\dot{u}^*(t)| \stackrel{\triangle}{=} |\dot{u}(v^*(t))| \leq \beta_2, \quad t \in \Omega_2.$$

Proof: See *Appendix A.14*.

Remark 5.4 *As can be seen from (5.41), the gradient of the hysteretic subsystem is highly nonlinear with respect to input $v(t)$ and output $u(t)$. By using only its bound information, the asymptotical error bound for σ_j in (5.46) may not be small enough, due to $\zeta_0/|q_h^0| \geq 2\lambda$. One more point is that the exact value of such bound is unknown to us beforehand. Therefore, the proposed dual-loop ILC could lose its efficiency in the case of large λ, and a better choice is to do search of gradient direction, in a lazy or intelligent way [154].*

Remark 5.5 *The dual-loop ILC design and convergence analysis under the second type of singularity can be done similarly as in the first singularity case or the normal cases.*

5.7 Illustrative Examples

Consider the tracking problem of the following dynamics

$$\begin{aligned} \dot{x} &= -2x + 3\sin\left(\frac{x}{2} + t\right) + u, \\ \dot{z} &= D^{-1}(A\dot{v} - \beta|\dot{v}||z|^{n-1}z - \gamma\dot{v}|z|^n), \\ u &= \alpha k v + (1-\alpha)Dkz, \quad z(0) = 0, \end{aligned} \tag{5.47}$$

where the unknown function $\eta(x,t) = -2x + 3\sin\left(\frac{x}{2} + t\right)$ satisfies the global Lipschitz condition with Lipschitz constant $L_\eta = 7/2$. The output $x(t)$ is desirable to track the reference signal $x_r(t), t \in [0, 10]$, which is determined by the reference controller input $v_r(t) = 2\sin t + \cos 2t - 1$ and its initial value $x_r(0) = 1$. In the simulation, the initial input signal is originated by a PD controller with appropriate gains ($k_p = 0.2, k_d = 0.01$), and the sampling time is set as 0.001 s.

Assume $n = 3, A = -1, D = 1, k = 5, \alpha = 0.55, \beta = 1$, and $\gamma = -2$, which satisfy the scenario (C_4'). Then, $\lambda = 5$ and $0 < q_h \leq 1/5$. Design the learning law as follows,

$$\begin{aligned} u_{i,r}(t) &= u_{i-1,r}(t) + 0.3e_i(t), \quad u_{0,r} = 0, \\ \Delta u_i(t) &= u_{i,r} - u(v_i, z(v_i, \dot{v}_i, t)), \\ v_i(t) &= v_{i-1}(t) + 0.2\Delta u_{i-1}(t). \end{aligned} \tag{5.48}$$

Figs. 5.20-5.23 give the simulation results. They reveal that the proposed dual learning control scheme can work well under hysteresis input uncertainty.

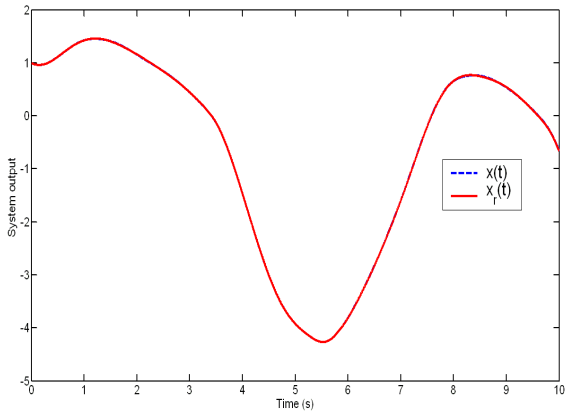


Figure 5.20: The learning result of system output $x(t), t \in [0, 10]$ with a stop condition $|e_i| < 0.01$. The reference trajectory $x_r(t)$ is determined by the whole system (5.1)-(5.3) with a desired input $v_r(t) = 2\sin t + \cos 2t - 1, t \in [0, 10]$.

Further simulate the scenario $\alpha = 0$. Set $n = 3, A = 0.1, D = 1, k = 5, \beta = 0.2, \gamma = 2.1$, and the desired input be $v_r(t) = \sin 2t + 10\cos t - 10, t \in [0, 10]$ for system (5.47) so that the desired hysteresis behavior can show obvious singularities, as shown in Fig. 5.19. This set of parameters belong to (C_2), and the corresponding $\lambda = 2kA = 1$. By (5.38), $0 < q_h < 1 - \zeta_0$. Choose $\zeta_0 = 0.02$ and $q_h = 0.2$ in (5.36). Figs. 5.24-5.27 give the simulation results, which also reflect the effectiveness of ILC strategy as some singularities involved.

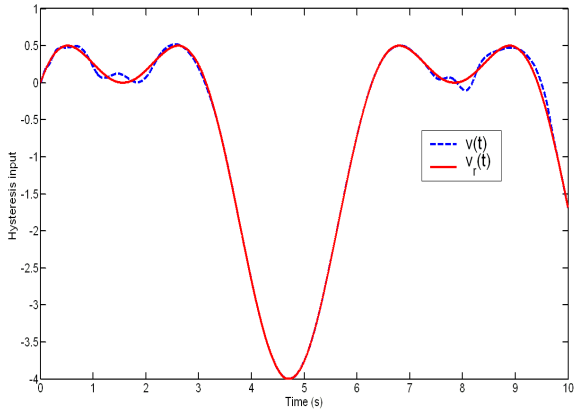


Figure 5.21: The learning result of the hysteresis input $v(t), t \in [0, 10]$.

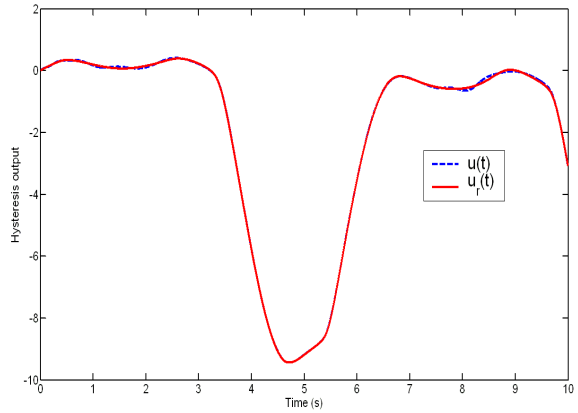


Figure 5.22: The learning result of the hysteresis output $u(t), t \in [0, 10]$. The reference trajectory $u_r(t)$ is given by the hysteresis part (5.2) and (5.3) with the desired input $v_r(t)$.

5.8 Conclusion

In this chapter, a dual-loop ILC scheme is designed for a class of nonlinear systems with hysteresis input uncertainty. The two ILC loops are applied to the nominal part and the hysteresis part respectively, to learn their unknown dynamics. Based on the convergence analysis for each single loop, a composite energy function method is then adopted to prove the learning convergence of the dual-loop system in iteration domain.

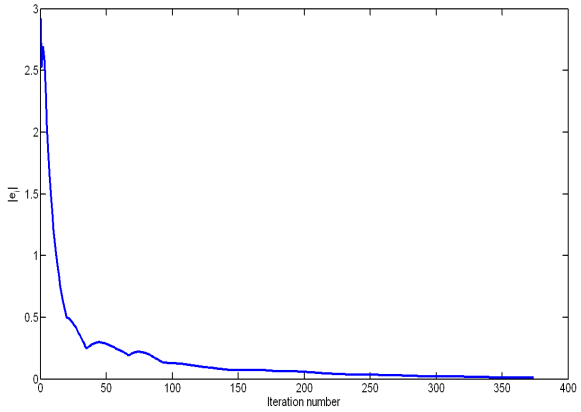


Figure 5.23: The variation of the maximal output error $|e_i|$ with respect to iteration number. Asymptotical convergence of tracking for systems with hysteretic input nonlinearity can be investigated with an acceptable error (≤ 0.01).

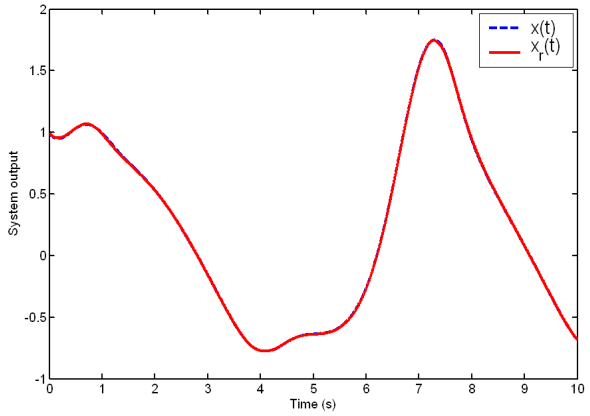


Figure 5.24: The learning result of system output $x(t), t \in [0, 10]$ with a stop condition $|e_i| < 0.01$ as $\alpha = 0$. The reference trajectory $x_r(t)$ is determined by the whole system (5.1)-(5.3) with a desired input $\sin 2t + 10 \cos t - 10, t \in [0, 10]$.

We further generalize the ILC law to deal with two singular cases in which the strict input-output monotonicity is violated. Note that the hysteretic output signal is assumed to be measurable in our control design. Our next aim is to consider more practical ILC design for the scenarios including that the internal hysteretic state is not accessible, and experiment test for those systems having hysteresis input uncertainties.

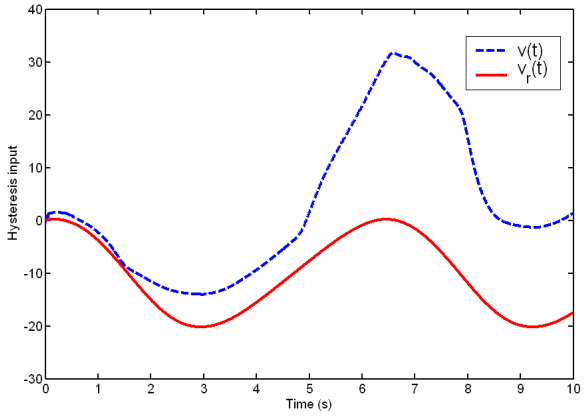


Figure 5.25: The learning result of the hysteresis input $v(t), t \in [0, 10]$ as $\alpha = 0$. It can be seen that the learned input signal $v(t)$, or the fixed-point input function $v^*(t)$ in the inner loop as $i \to \infty$ could show much deviation compared with the desired input $v_r(t)$. Even so, they will yield similar hysteretic output profiles. Investigating the hysteresis dynamics as $\alpha = 0$, $\dot{u} = \dot{}(kA - |u|^n/(k^{n-1}D^n)(\gamma + \beta\mathcal{S}(\dot{v}u)))$, the hysteretic output $u(t)$ is relevant to $\dot{v}$ and its sign if the factor $kA - |u|^n/(k^{n-1}D^n)(\gamma + \beta\mathcal{S}(\dot{v}u))$ does not vanish, and otherwise relevant to its sign only.

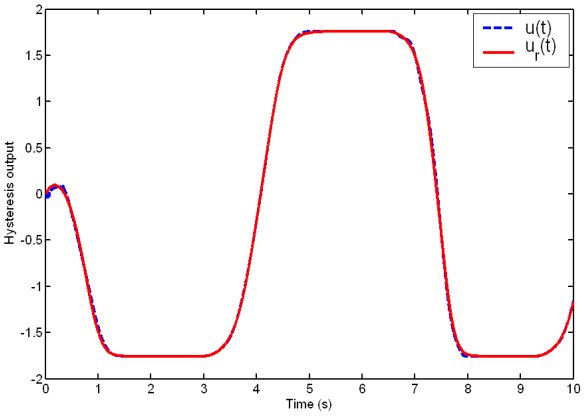


Figure 5.26: The learning result of the hysteresis output $u(t), t \in [0, 10]$ as $\alpha = 0$. The reference trajectory $u_r(t)$ is given by the hysteresis part (5.2) and (5.3) with the desired input $v_r(t) = \sin 2t + 10\cos t - 10, t \in [0, 10]$.

Compared with the process control governed by ODEs, the control of PDEs will illustrate more different characteristics. In the next phase, we consider the boundary

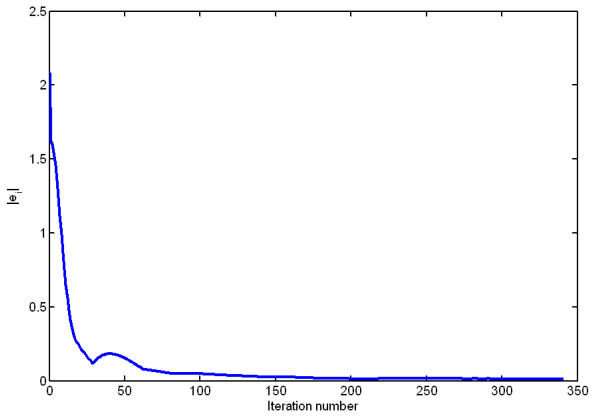


Figure 5.27: The variation of the maximal output error $|e_i|$ with respect to iteration number as $\alpha = 0$. Asymptotical convergence of tracking for systems with hysteretic input nonlinearity can be investigated with an acceptable error (≤ 0.01).

control for a class of strictly repeatable biochemical processes, by adopting the idea of ILC.

Chapter 6

Iterative Boundary Learning Control for a Class of Nonlinear PDE Processes

6.1 Introduction

Compared with the control of processes described by ODEs, fewer control schemes have been developed for processes described by PDEs. A major portion of established PDE control schemes focus on the use of distributed actuation, namely, the control action depends on the spatial coordinates. However, in many important industrial processes the control actuation is achieved through the boundary of the process, such as the case of chemical and biochemical reactors where the manipulated input is the fluid velocity at the feed of the process [37, 63]. In [107]- [110], the boundary control of PDEs with adaptive control methodology is extended to cope with either stable or unstable PDEs. These works are built upon explicitly parameterized control formulae to avoid solving Riccati or Bezout equations at each time step. Backstepping is also adopted to solve the problem of stabilization of some PDEs by using boundary control in [69, 111]. In practice, however, simple controllers such as PI or PID compensators are most widely

used by process engineers in the chemical and biochemical industry, owing to many reasons such as implementability, the long history of proven operation and robustness, and the fact that these simple controllers are well understood by industrial practitioners.

We know that the major difficulty in PDE control is how to optimally tune the controller gains. When process uncertainties are present, it is almost impossible to find the values or bounds of the controller gains such that the closed-loop performance can be guaranteed for the PDE processes, as can be seen from [63].

In this chapter, we assume that the PDE procuress under consideration is strictly repeatable, which is one of the main features in certain types of real process control including industrial chemical [34] and biochemical reactors [37]. We develop an ILC for a class of SISO nonlinear PDE processes with boundary control, and in the presence of parametric/non-parametric uncertaitnies affecting the interior of the domain. The control objective is to iteratively tune the velocity boundary condition on one side such that the boundary output on the other side can be regulated to a desired level. The main result of the chapter shows that, under physically reasonable input-output monotonicity, stability and steady-state assumptions, the desired regulation output can be achieved with an acceptable small error by iteratively tuning the boundary velocity in finite iterations, under input saturation and system uncertainties, where the feasible bound of the IBLC gain and its learning rate are clearly analyzed.

This chapter is organized as follows. Section 6.2 presents the PDE systems and state the control problem. Section 6.3 presents the main result. Section 6.4 gives an example on a tubular bioreactor. Section 6.5 closes the work with some conclusions.

6.2 System Description and Problem Statement

Consider the following SISO quasi-linear PDE processes with velocity boundary control [63]

$$\begin{aligned}\frac{\partial \mathbf{c}(z,t)}{\partial t} &= -B\frac{\partial (v(z,t)\mathbf{c}(z,t))}{\partial z} + D\frac{\partial^2 \mathbf{c}(z,t)}{\partial z^2} \\ &\quad + \mathbf{f}(\mathbf{c}(z,t),z), \qquad (6.1)\end{aligned}$$

$$\frac{\partial v(z,t)}{\partial t} = -v(z,t)\frac{\partial v(z,t)}{\partial z}, \qquad (6.2)$$

with the controlled output

$$y = h(\mathbf{c}(L,t)), \qquad (6.3)$$

the boundary conditions

$$A_1\mathbf{c}(0,t) + B_1\frac{\partial \mathbf{c}(0,t)}{\partial z} = C_1, \qquad (6.4)$$

$$A_2\mathbf{c}(L,t) + B_2\frac{\partial \mathbf{c}(L,t)}{\partial z} = C_2, \qquad (6.5)$$

and

$$a_1 u(t) + b_1\frac{\partial v(0,t)}{\partial z} = c_1, \qquad (6.6)$$

$$a_2 v(L,t) + b_2\frac{\partial v(L,t)}{\partial z} = c_2, \qquad (6.7)$$

and the initial condition

$$\mathbf{c}(z,0) = \mathbf{c}_0(z), \qquad (6.8)$$

$$v(z,0) = v_0(z). \qquad (6.9)$$

Here, $\mathbf{c} \in \mathcal{H}([0,L]\times[0,T],\mathbf{R}^n)$ is the vector of process variables, $v \in \mathcal{H}([0,L]\times[0,T],\mathbf{R})$ is the fluid velocity where $0 < L,T < \infty$, and $y = h(\mathbf{c}) \in \mathbf{R}$ denotes the controlled output, the boundary condition $u(t) = v(0,t) \in \mathbf{R}$ denotes the manipulated variable,

$z \in [0, L]$ is the spatial coordinate, $t \in [0, T]$ is the time. $A_i, B_i, i = 1, 2, B > 0$ and $D > 0$ are matrices of suitable dimension, and $a_i, b_i, c_i, i = 1, 2$ are scalar parameters. In Eqs. (6.1) and (6.3), $\mathbf{f}(\mathbf{c}, z)$ and $h(\mathbf{c})$ are C^1 nonlinear functions, satisfying the Lipschitz conditions, i.e., for $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{H}([0, L]\times[0, T], \mathbf{R}^n)$, $z \in [0, L]$, there exist a known integrable Lipschitz function $\omega_{\mathbf{f}}(z)$ and another Lipschitz constant ω_h such that

$$\|\mathbf{f}(\mathbf{c}_1, z) - \mathbf{f}(\mathbf{c}_2, z)\| \leq \omega_{\mathbf{f}}(z)\|\mathbf{c}_1 - \mathbf{c}_2\|, \tag{6.10}$$

and

$$\|h(\mathbf{c}_1) - h(\mathbf{c}_2)\| \leq \omega_h\|\mathbf{c}_1 - \mathbf{c}_2\|. \tag{6.11}$$

The above PDE models generally describe the diffusion-convection phenomena in some open-loop processes. Many important industrial processes can be formulated within this modelling framework, e.g., industrial chemical [37], biochemical reactors [34], heat exchangers [98], and biofilters for air and water pollution control [106]. For instance, when regulating the total amount of output flow pollutions (e.g. toluene vapor) via manipulations of the input flow rate in an airstream biofilter, the vector $\mathbf{c}(z, t)$ denotes the distribution of pollutants, the parameters B and D are the convection and diffusion coefficients of the pollutions in filter, and the term $\mathbf{f}(\mathbf{c}(z, t), z)$ represents the bio-reaction rate that affects the pollution concentration in filter. On the other hand, Eq. (6.2) approximates the fluid velocity field along the process, and it keeps the basic feature for our control design and analysis. When the fluid is incompressible, it can be written as $v(z, t) = u(t)$ directly. Actually, the plant (6.2) reveals that the disturbances in the feed velocity $u(t)$ are transported along the process, and it would induce certain delay in the control action. Notice that even if $\mathbf{f}(\mathbf{c}(z, t), z)$ is a linear function and the fluid is incompressible, the control problem is still nonlinear due to the product $v\partial\mathbf{c}/\partial z$. Our

discussion in the Section III states that the proposed IBLC scheme is also applicable when more sophisticated velocity field models are adopted.

For the PDE process (6.1)-(6.9), two restrictions or assumptions are usually involved: velocity saturation and output measurement ability. The details are as follows:

Assumption 6.1 *The process is operated within the velocity restriction* $v(z,t) \in [v_{min},\ v_{max}]$, *with* $v_{max} > v_{min} \geq 0$.

Assumption 6.2 *The controlled output* $y = h(\mathbf{c})$ *is available for measurements with certain time delay.*

Remark 6.1 *From the physical point of view, Assumptions 6.1 and 6.2 are fundamental and rational. The output measurement ability enables us to use the feedback-type control scheme while the existence of velocity bounds facilitate the convergence design by choosing appropriate learning gains. Since only a rough estimation of the velocity bounds is needed here, it can be available to us before implementing the proposed control law. Moreover, since what we concern is in the steady-state period only, any finite measurement delay would not degrade the control performance.*

In this work, we also assume the PDE process is strictly repeatable in time domain and spatial domain so that the ILC scheme can be utilized here in an iterative manner. This kind of processes may be frequently encountered in chemical or biochemical industry. Let $y^* \in \mathbf{R}$ be any given set point that can be regulated for the process output by tuning the boundary velocity $u(t)$. The control task is to design a simple ILC law

$$u_{i+1}(t) = u_i(t) + \rho(y^* - y_i(t)), \tag{6.12}$$

where i is the iteration number, such that the regulation error $y^* - y_i(t)$ can converge into an acceptable neighborhood of zero pointwisely within finite iterations. ρ is the undetermined learning gain.

Let

$$F_1(\mathbf{c}, v) \triangleq -B\frac{\partial(v\mathbf{c})}{\partial z} + D\frac{\partial^2 \mathbf{c}}{\partial z^2} + \mathbf{f}(\mathbf{c}, z), \tag{6.13}$$

$$F_2(v) \triangleq -v\frac{\partial v}{\partial z}. \tag{6.14}$$

Consider a constant boundary velocity input $u(t) = \bar{u} > 0, t \in [0, T]$. Then, the steady state velocity, defined by $\bar{v}(z)$, satisfies $F_2(\bar{v}(z)) = 0$, or equivalently, $\bar{v}(z) = 0$ or $\partial\bar{v}/\partial z = 0$. Since $\bar{v}(0) = \bar{u} > 0$, it yields that

$$\bar{v}(z) = \bar{u}, \quad z \in [0, L]. \tag{6.15}$$

Thus, the corresponding steady state of the process, denoted by $\bar{\mathbf{c}}$, satisfies the relationship $F_1(\bar{\mathbf{c}}, \bar{u}) = 0$. If $\bar{\mathbf{c}}$ is uniquely determined by the implicit function $F_1(\bar{\mathbf{c}}, \bar{u}) = 0$, it should be more rational to revise the IBLC law as follows

$$\bar{u}_{i+1} = \bar{u}_i + \rho(y^* - \bar{y}_i), \quad \bar{y}_i \triangleq h(\bar{\mathbf{c}}(\bar{u}_i)), \tag{6.16}$$

since the output set-point problem at the steady state stage is considered only. Therefore, two more assumptions are given as follows to make valid of our subsequent discussion.

Assumption 6.3 *Each steady state output $\bar{y} \triangleq h(\bar{\mathbf{c}}(\bar{u}))$ is achieved by one and only one constant boundary velocity input $\bar{u}$.*

Remark 6.2 *The above assumption implies that the map $\bar{u} \to h(\bar{\mathbf{c}}(\bar{u}))$ is a bijection map, or equivalently the strict monotonicity of input-output relationship holds, i.e.,*

$$\left(\frac{\partial h}{\partial \mathbf{c}}\right)^T \frac{\partial \mathbf{c}}{\partial \bar{u}} > (<)0, \ \forall \bar{u} \in [v_{min}, v_{max}].$$

This is physically reasonable, and many control processes possess such a good property, e.g., the anaerobic digestion process mentioned below where the steady-state output $\bar{y} = \bar{c}(L)$ *is a strictly increasing function of the steady-state velocity input* $\bar{u}$*. This condition implies the fact that the larger the feed velocity, the smaller the residence time of the pollutants, which yields lower degradation of pollutants.*

Of course, a general application of ILC may not need the injectivity of input-output map. As such, the desired input signal could become non-unique and the contraction mapping method we used here will lose its efficiency.

Next, one more stability and steady state assumption is given.

Assumption 6.4 *For all constant input* $\bar{u} \in [v_{min}, v_{max}]$*, and all initial condition* $c_0(z) \in \mathcal{H}([0, L], \mathbf{R}^n), v_0(z) \in \mathcal{H}([0, L], \mathbf{R})$*, the solution of (6.1)-(6.9) is uniformly asymptotically stable (uniform in the constant input* $\bar{u}$*), namely, the process will reach steady state after a sufficient time interval.*

Remark 6.3 *The above stability assumption implies that the energy of the perturbation variables* $\mathbf{c}(z,t) - \bar{\mathbf{c}}(z)$ *and* $v(z,t) - \bar{v}(z)$ *decays in an asymptotical way at any position* z*. In a qualitative point of view, it states the diffusion-convection ability, source effect, and boundary condition effect for the process. Actually, many real processes show their exponential stability in time, and all kinds of methods, e.g., energy methods [113], are utilized to detect the decay rate in analysis. The process example we presented in Section IV was proven to satisfy the exponential stability in [63], and thus the asymptotical stability assumption.*

Based on *Assumption* 6.4, without loss of generality, we further assume that for any

$\epsilon > 0$ there exists $T_\epsilon < T$ such that $\max\{\|\mathbf{c}(z,t) - \bar{\mathbf{c}}(z)\|, |v(z,t) - \bar{v}(z)|\} < \epsilon$ for all $t \in [T_\epsilon, T], \bar{u} \in [v_{min}, v_{max}]$.

6.3 IBLC for the Nonlinear PDE Processes

In this section, we first analyze the convergence of the IBLC algorithm, and then give an estimation for its learning rate. In the end, a more complex fluid velocity dynamics is addressed.

6.3.1 Convergence of the IBLC

Recall that gradient estimation of the input-output mapping is always crucial for any ILC design, and its precise information should be helpful for us to design the learning gain such that fast control convergence can be derived. Note that *Assumptions* 6.3 and 6.4 enable us to consider the IBLC at steady state stage only. As the first step, we aim to derive a quantitative bound for the steady state of process $\bar{\mathbf{c}}(z)$, whose existence has been implied by *Assumption* 6.4, and it can be used to estimate the gradient in the following.

Property 6.1 *Under the above assumptions, the steady state of process* $\bar{\mathbf{c}}(z)$ *is bounded from the above by the quantity*

$$\begin{aligned} & \Xi_0(z) \\ \triangleq\ & \left(\left(1 + v_{max} z \left\|D^{-1}B\right\|\right) \|\bar{\mathbf{c}}(0)\| + z \left\| \frac{\partial \bar{\mathbf{c}}(0)}{\partial z} \right\| \right) \\ & \times \exp\left(\int_0^z \left[v_{max} \|D^{-1}B\| \right.\right. \\ & \left.\left. + \|D^{-1}\|(z-\tau)\omega_{\mathbf{f}}(\tau) \right] d\tau \right). \end{aligned}$$

Proof: See *Appendix A.15*.

With the help of Property 6.1, we can achieve our main result.

Theorem 6.1 *Consider the PDE process (6.1)-(6.9) satisfying Assumptions 6.1-6.4 and under the ILC law (6.16). If* $\bar{u}^* \in [v_{min}, v_{max}]$, *where* $\bar{u}^*$ *is the desired constant boundary velocity corresponding to* y^*, *and* $sign(\rho) = sign\left(\left(\frac{\partial h}{\partial \bar{\mathbf{c}}}\right)^T \frac{\partial \bar{\mathbf{c}}}{\partial \bar{u}}\right)$ *and*

$$\frac{1-\delta}{\lambda} < |\rho| < \frac{1+\delta}{\lambda}, \quad 0 < \delta < 1, \tag{6.17}$$

where $\lambda = \omega_h \Xi_1$ *with*

$$\begin{aligned} \Xi_1 &= \|D^{-1}B\| \left(\int_0^L \|\Xi_0(z)\| dz + \|\bar{\mathbf{c}}^*(0)\| L \right) \\ &\quad \times \exp\left(\int_0^L \left(v_{max} \|D^{-1}B\| \right.\right. \\ &\quad \left.\left. + \|D^{-1}\| (L-z) \omega_{\mathbf{f}}(z) \right) dz \right), \end{aligned} \tag{6.18}$$

then for any $\epsilon > 0$ *given in Assumption 6.4,* $|y_i(t) - y^*| \leq \omega_h \epsilon$ *as* $i \to \infty$ *and* $t \in [T_\epsilon, T]$.

Proof: See *Appendix A.16*.

In real process control, input saturation is another concern we have to address. Following the IBLC law (6.16), the control input signal may become not implementable. With the saturation a control process becomes highly nonlinear, even for linear time-invariant systems. The existence of input saturation might cause the exhibition of a limit cycle or even unstable performance [143]. Therefore, the analysis of an iterative learning control process with input saturation would be very meaningful and indispensable in practice. Next, we will show that the IBLC with input saturation can be discussed in a similar way as the original IBLC without any input saturation.

Revise (6.16) to the actual case

$$\bar{u}_{i+1} = \mathrm{Proj}\left(\bar{u}_i + \rho(y^* - \bar{y}_i)\right), \quad \bar{y}_i \stackrel{\triangle}{=} h(\bar{\mathbf{c}}(\bar{u}_i)), \tag{6.19}$$

where the projection operator Proj(·) is defined as follows.

$$\text{Proj}(\bar{u}) = \begin{cases} v_{min}, & \bar{u} \leq v_{min}, \\ v_{max}, & \bar{u} \geq v_{max}, \\ \bar{u}, & \text{otherwise.} \end{cases} \tag{6.20}$$

Property 6.2 *For a given $\bar{u}^* \in [v_{min}, v_{max}]$, the following inequality holds:*

$$|\bar{u}^* - Proj(\bar{u})| \leq |\bar{u}^* - \bar{u}|. \tag{6.21}$$

Proof. Note that $v_{min} \leq \bar{u}^* \leq v_{max}$. If $\bar{u} \geq v_{max}$, then $\text{Proj}(\bar{u}) = v_{max} \leq \bar{u}$, and subsequently $0 \leq v_{max} - \bar{u}^* = \text{Proj}(\bar{u}) - \bar{u}^* \leq \bar{u} - \bar{u}^*$, implying (6.21). If $\bar{u} \leq v_{min}$, then $\text{Proj}(\bar{u}) = v_{min} \geq \bar{u}$, and subsequently $0 \geq v_{min} - \bar{u}^* = \text{Proj}(\bar{u}) - \bar{u}^* \geq \bar{u} - \bar{u}^*$, also implying (6.21). If $\bar{u} \in [v_{min}, v_{max}]$, then $\text{Proj}(\bar{u}) = \bar{u}$ and $|\bar{u}^* - \text{Proj}(\bar{u})| = |\bar{u}^* - \bar{u}|$. ■

According to Property 6.2, the convergence of input error for (6.19) can be derived as follows.

$$\begin{aligned} |\Delta\bar{u}_{i+1}| &= |\bar{u}^* - \bar{u}_{i+1}| \\ &= |\bar{u}^* - \text{Proj}\left(\bar{u}_i + \rho(y^* - \bar{y}_i)\right)| \\ &\leq |\bar{u}^* - \bar{u}_i - \rho(y^* - \bar{y}_i)|. \end{aligned} \tag{6.22}$$

Then, similar to the discussion from (8.103) to (8.105), we have that

$$|\Delta\bar{u}_{i+1}| \leq \delta|\Delta\bar{u}_i| < |\Delta\bar{u}_i|,$$

where the relationships (6.17) and (8.100) are used.

6.3.2 Learning rate evaluation

Since the parameter ϵ could be arbitrarily small in Theorem 6.1, the output regulation error $y_i(t) - y^*$ in the IBLC can converge into a sufficiently small neighborhood of zero.

However, this is due to the infinite iterations or experimental trials which may not be feasible in real implementation. To evaluate the performance of proposed scheme in finite iterations, we ignore the steady-state estimation error, that is, assume $\epsilon = 0$, and present the following result.

Theorem 6.2 *Let all the notations be same as in Theorem 6.1. For any given $\epsilon_1 > 0$, by applying the control law (6.16) and choosing the learning gain in the range as given in (6.17), the output $\bar{y}_i$ will converge to the ϵ_1-neighborhood of the desired set-point y^* with a finite number of iterations no more than*

$$N_{\epsilon_1} = \frac{\log \frac{\epsilon_1}{(v_{max} - v_{min})\lambda}}{\log\left(1 - (1-\delta)\frac{\epsilon_1}{(v_{max} - v_{min})\lambda}\right)} + 1.$$

where λ is given below (6.17).

Proof: See *Appendix A.17*.

Remark 6.4 *After N_{ϵ_1} iterations, the actual output regulation error should be bounded as follows:*

$$\begin{aligned} |y_i(t) - y^*| &\leq |\Delta\bar{y}_i| + |y_i(t) - \bar{y}_i| \\ &\leq \epsilon_1 + \omega_h \epsilon, \quad t \in [T_\epsilon, T]. \end{aligned}$$

6.3.3 Extension to more general fluid velocity dynamics

Eq. (6.2) represents an approximate model of the fluid velocity field along the process without fluid dispersion. Its structure simplifies our IBLC convergence proof because $\bar{v}(z) = \bar{u}$ is a constant along the spatial coordinate z. Actually, more sophisticated

velocity dynamics can be considered in a similar way. Next, assume the velocity part takes the following structure

$$\frac{\partial v(z,t)}{\partial t} = -B_v \frac{\partial v(z,t)}{\partial z} + D_v \frac{\partial^2 v(z,t)}{\partial z^2} \tag{6.23}$$

with boundary conditions

$$v(0,t) = u(t), \tag{6.24}$$

$$a_2 v(L,t) + b_2 \frac{\partial v(L,t)}{\partial z} = c_2, \tag{6.25}$$

where the parameters B_v, D_v, a_2, b_2, and c_2 are non-negative constants. For this case, the steady state of $v(z,t)$, determined by equation

$$F_2(\bar{v}, \bar{u}) \triangleq -B_v \frac{\partial \bar{v}(z)}{\partial z} + D_v \frac{\partial^2 \bar{v}(z)}{\partial z^2} = 0, \tag{6.26}$$

is not of constant any more but a function of the constant boundary velocity $\bar{u}$ and spatial coordinate z. Specifically, solving (6.26) with boundary conditions $\bar{v}(0) = \bar{u}, a_2\bar{v}(L) + b_2 d\bar{v}(L)/dz = c_2$ yields that

$$\begin{aligned} &\bar{v}(z,\bar{u}) \\ = &\frac{-c_2 D_v + \bar{u}(a_2 D_v + b_2 B_v)\exp\left(\frac{B_v}{D_v}L\right)}{(a_2 D_v + b_2 B_v)\exp\left(\frac{B_v}{D_v}L\right) - a_2 D_v} \\ &- \frac{D_v(a_2\bar{u} - c_2)}{(a_2 D_v + b_2 B_v)\exp\left(\frac{B_v}{D_v}L\right) - a_2 D_v} \\ &\times \exp\left(\frac{B_v z}{D_v}\right), \end{aligned} \tag{6.27}$$

whose derivative with respect to $\bar{u}$ is

$$\frac{(a_2 D_v + b_2 B_v)\exp\left(\frac{B_v}{D_v}L\right) - a_2 D_v \exp\left(\frac{B_v}{D_v}z\right)}{(a_2 D_v + b_2 B_v)\exp\left(\frac{B_v}{D_v}L\right) - a_2 D_v}. \tag{6.28}$$

Noticing that $0 \le z \le L$ and the non-negativity of parameters B_v, D_v, a_2, b_2, and c_2, we can see that

$$0 \le \frac{b_2 B_v \exp\left(\frac{B_v}{D_v}L\right)}{(a_2 D_v + b_2 B_v)\exp\left(\frac{B_v}{D_v}L\right) - a_2 D_v} \le \frac{\partial \bar{v}}{\partial \bar{u}},$$

and

$$\kappa := \frac{(a_2 D_v + b_2 B_v)\exp\left(\frac{B_v}{D_v}L\right)}{(a_2 D_v + b_2 B_v)\exp\left(\frac{B_v}{D_v}L\right) - a_2 D_v} \geq \frac{\partial \bar{v}}{\partial \bar{u}}. \tag{6.29}$$

Thus, the bounds and sign information of the gradient $\partial\bar{v}/\partial\bar{u}$ are known to us, and then the IBLC for this scenario can be analyzed along the way we presented before.

Integrating $F_1(\bar{\mathbf{c}}(z), \bar{v}(z)) = \mathbf{0}$, where F_1 is defined in (6.13), along the spatial coordinate from 0 to z,

$$\begin{aligned} &-B\bar{v}(z)\bar{\mathbf{c}}(z) + D\frac{\partial \bar{\mathbf{c}}(z)}{\partial z} + \left(B\bar{v}(0)\bar{\mathbf{c}}(0) - D\frac{\partial \bar{\mathbf{c}}(0)}{\partial z}\right) \\ &+ \int_0^z \mathbf{f}(\bar{\mathbf{c}}(\tau), \tau)d\tau = \mathbf{0}. \end{aligned} \tag{6.30}$$

Further integrating (6.30), it yields that

$$\begin{aligned} &-B\int_0^z \bar{v}(\tau)\bar{\mathbf{c}}(\tau)d\tau + D(\bar{\mathbf{c}}(z) - \bar{\mathbf{c}}(0)) \\ &+ \left(B\bar{v}(0)\bar{\mathbf{c}}(0) - D\frac{\partial \bar{\mathbf{c}}(0)}{\partial z}\right) z \\ &+ \int_0^z (z-\tau)\mathbf{f}(\bar{\mathbf{c}}(\tau), \tau)d\tau = \mathbf{0}, \end{aligned} \tag{6.31}$$

or equivalently

$$\begin{aligned} \bar{\mathbf{c}}(z) \;=\; & \bar{\mathbf{c}}(0) + D^{-1}B\int_0^z \bar{v}(\tau)\bar{\mathbf{c}}(\tau)d\tau \\ & - \left(D^{-1}B\bar{v}(0)\bar{\mathbf{c}}(0) - \frac{\partial \bar{\mathbf{c}}(0)}{\partial z}\right) z \\ & - D^{-1}\int_0^z (z-\tau)\mathbf{f}(\bar{\mathbf{c}}(\tau), \tau)d\tau. \end{aligned} \tag{6.32}$$

Noticing the velocity restriction $\bar{v}(z) \in [v_{min}, v_{max}]$, the *Property* 6.1, i.e. $\|\bar{\mathbf{c}}(z)\| \leq \Xi_0(z)$, still holds for this scenario by applying the generalized Gronwall inequality.

Now, we are in the position of deriving the relationship of input/output errors by using the formula (6.32). Similarly as from (8.97) to (8.99), using the strict repeatable assumption for the process, the boundedness property of $\bar{\mathbf{c}}(z)$, the velocity restriction,

and the Lipschitz condition of $\mathbf{f}(\bar{\mathbf{c}}(z), z)$ in i-th iteration will give that

$$
\begin{aligned}
& \|\Delta\bar{\mathbf{c}}_i(L)\| \\
= \; & \Big\| D^{-1}B\left(\int_0^L \bar{v}^*(z)\bar{\mathbf{c}}^*(z)dz - \bar{v}^*(0)\bar{\mathbf{c}}^*(0)L\right) \\
& - D^{-1}\int_0^L (L-z)\mathbf{f}(\bar{\mathbf{c}}^*(z), z)dz \\
& -D^{-1}B\left(\int_0^L \bar{v}_i(z)\bar{\mathbf{c}}_i(z)dz - \bar{v}_i(0)\bar{\mathbf{c}}_i(0)L\right) \\
& +D^{-1}\int_0^L (L-z)\mathbf{f}(\bar{\mathbf{c}}_i(z), z)dz\Big\| \\
\leq \; & \|D^{-1}B\|\|\bar{v}_i(0)\bar{\mathbf{c}}_i(0) - \bar{v}^*(0)\bar{\mathbf{c}}^*(0)\|L \\
& + \|D^{-1}B\|\int_0^L |\bar{v}^*(z)|\|\Delta\bar{\mathbf{c}}_i(z)\|dz \\
& + \|D^{-1}B\|\int_0^L |\bar{v}^*(z) - \bar{v}_i(z)|\|\bar{\mathbf{c}}_i(z)\|dz \qquad (6.33) \\
& + \|D^{-1}\|\int_0^L (L-z) \\
& \quad \times \|\mathbf{f}(\bar{\mathbf{c}}^*(z), z) - \mathbf{f}(\bar{\mathbf{c}}_i(z), z)\|dz \\
\leq \; & \|D^{-1}B\|\|\bar{\mathbf{c}}^*(0)\|L|\bar{v}^*(0) - \bar{v}_i(0)| \\
& + \|D^{-1}B\|\int_0^L |\bar{v}^*(z) - \bar{v}_i(z)|\Xi_0(z)dz \\
& + \int_0^L \left(v_{max}\|D^{-1}B\| + \|D^{-1}\|(L-z)\omega_{\mathbf{f}}(z)\right) \\
& \quad \times \|\Delta\bar{\mathbf{c}}_i(z)\|dz. \qquad (6.34)
\end{aligned}
$$

In (6.34), the velocity error $\bar{v}^*(z) - \bar{v}_i(z)$ is relevant to the input error $\Delta\bar{u}_i$. By the Mean Value Theorem, there exists a point η_i between $\bar{u}^*$ and $\bar{u}_i$ such that for $z \in [0, L]$

$$
|\bar{v}^*(z) - \bar{v}_i(z)| \leq \left|\frac{\partial\bar{v}}{\partial\bar{u}}(z, \eta_i)\right| |\Delta\bar{u}_i| \leq \kappa|\Delta\bar{u}_i|, \qquad (6.35)
$$

where κ, defined in (6.29), is the upper bound of $\partial\bar{v}/\partial\bar{u}$. Subsequently, the relationship between input error and output error is determined by the following inequality

$$
\|\Delta\bar{\mathbf{c}}_i(L)\|
$$

$$\begin{aligned} \leq \quad & \kappa\|D^{-1}B\| \left(\int_0^L \Xi_0(z)dz + \|\bar{\mathbf{c}}^*(0)\|L \right) |\Delta\bar{u}_i| \\ & + \int_0^L \left(v_{max}\|D^{-1}B\| + \|D^{-1}\|(L-z)\omega_{\mathbf{f}}(z) \right) \\ & \times \|\Delta\bar{\mathbf{c}}_i(z)\|dz. \end{aligned} \tag{6.36}$$

Similar to the proof in *Theorem* 6.1, it yields by using the generalized Gronwall inequality that

$$\|\Delta\bar{\mathbf{c}}_i(L)\| \leq \Xi_1'|\Delta\bar{u}_i|, \tag{6.37}$$

where $\Xi_1' = \kappa\Xi_1$.

Now, substituting Ξ_1 by Ξ_1' in *Theorems* 6.1 and 6.2, the remaining part of the IBLC convergence proof and its learning rate evaluation for this scenario can be done in a same way as in the first two parts of this section.

6.4 Illustrative Example and Its Simulation

In this section, we consider an anaerobic digestion process control for wastewater treatment by using the proposed IBLC scheme. The anaerobic digestion takes places in a fluidized-like bio-reactor where biomass (bacteria consortium) is attached on a plastic support. Once after biomass attains a stable structure and concentration, wastewater is fed to the bio-reactor, in which pollutants are converted into methane and carbon dioxide gases [25, 63]. The process model is expressed as follows:

$$\begin{aligned} \frac{\partial c(z,t)}{\partial t} \quad = \quad & -B\frac{\partial(v(z,t)c(z,t))}{\partial z} + D\frac{\partial^2 c(z,t)}{\partial z^2} \\ & - \mu c(z,t), \end{aligned} \tag{6.38}$$

$$\frac{\partial v(z,t)}{\partial t} \quad = \quad -v(z,t)\frac{\partial v(z,t)}{\partial z} \tag{6.39}$$

with the controlled output $y = c(L,t)$, the initial conditions $c(z,0) = c_0, v(z,0) = v_0$, and the boundary conditions:

$$c(0,t) = C_1, \quad \frac{\partial c(L,t)}{\partial z} = 0, \tag{6.40}$$

$$v(0,t) = u(t) = \bar{u}, \ \frac{\partial v(L,t)}{\partial z} = 0. \tag{6.41}$$

In this model, the non-negative states $c(z,t), v(z,t) \in \mathbf{R}, z \in [0,L], t \in [0,T]$ denote the pollutant concentration in the wastewater and the feed flow rate respectively, $f(c) = -\mu c$ is the bio-reaction rate at which pollutants are converted into bio-gas for the scenario of low and moderate concentrations, B and D are the parameters relevant to advection average velocity and dispersion respectively. The control task is to keep the effluent pollutant concentration $y(t)$ at a given constant value y^* by tuning the feed flow rate $u(t)$ at the boundary $z = 0$. Efficient regulation of $y(t)$ is very important for environment conservation in both industrial and municipal wastewaters [119]. For the case of $v(z,t) = v(t)$, [63] has considered this problem by using PI-type controller. Next, we will show that the proposed IBLC controller is efficient to achieve effluent regulation under repeatable environment. For illustration, let $L = 1.0\ m, T = 48\ h, B = 1.0\ mg^{-1}h^{-1}, D = 0.1\ m^2g^{-1}h^{-1}, \mu = 0.17\ h^{-1}, v_{min} = 0.05\ mh^{-1}, v_{max} = 0.8\ mh^{-1}, C_1 = 1.15\ gl^{-1}, c_0 = 0, v_0 = 0.25\ mh^{-1}$, $y^* = 0.3\ gl^{-1}$, $\epsilon = 0.001$, and $\delta = 0.8$. With these values we can calculate that $\lambda = 26.17$, and $0.008 < |\rho| < 0.068$ by (6.17).

At the steady state stage, the pollutant concentration in the wastewater $\bar{c}(z)$ satisfies the following equation

$$D\frac{d^2\bar{c}(z)}{dz^2} - B\bar{u}\frac{d\bar{c}(z)}{dz} - \mu\bar{c} = 0, \tag{6.42}$$

with its boundary conditions $\bar{c}(0) = C_1$ and $d\bar{c}(L)/dz = 0$. For this linear system and

its PDE correspondence (6.38)-(6.41), it is easy to check that Assumptions 6.3 and 6.4 are satisfied, where $d\bar{y}/d\bar{u} > 0$ for $v_{min} \leq \bar{u} \leq v_{max}$, as can be seen in [63]. Setting the learning gain ρ be 0.06 and starting with $\bar{u}_0 = 0.75\ mh^{-1}$, Figs. 6.1 and 6.2 give the steady output error and constant input profiles respectively. With the learned optimal input $u(t) = \bar{u} = 0.1232\ mh^{-1}$, Figs. 6.3 and 6.4 show the variations of the pollutant concentration $c(z,t)$ and the feed flow rate $v(z,t)$ in three dimensional space.

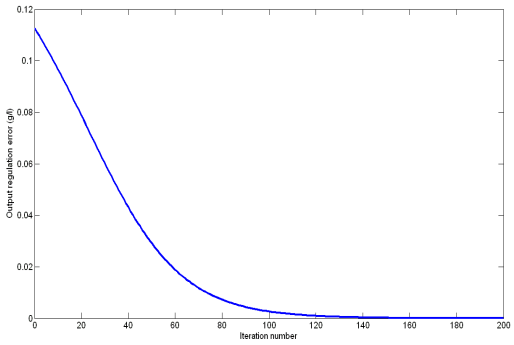


Figure 6.1: Output regulation error profile by using the proposed IBLC controller with $\rho = 0.06$. It can be seen that the output regulation achieves the desired set-point after around 140 iterations.

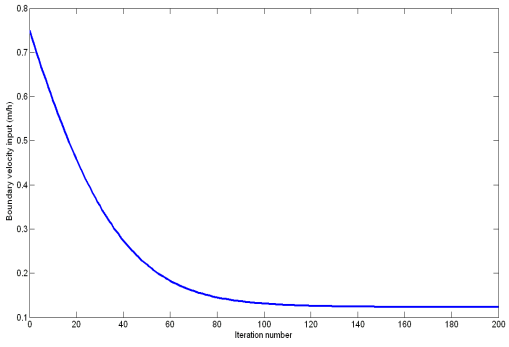


Figure 6.2: Constant boundary velocity input profile updated by the IBLC law. The desired constant input is 0.1232 mh^{-1}. During all the iterations, control inputs always lie in the saturation bound $[0.05, 0.8]$.

Obviously, the gain constraint (6.17) is fully determined by the value of parameter λ,

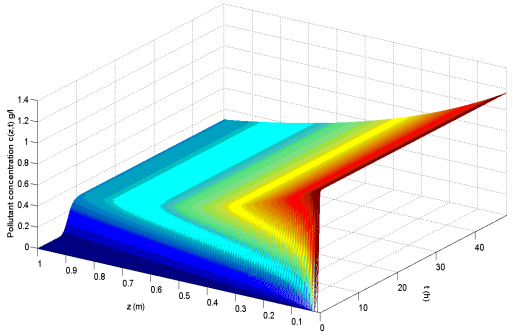


Figure 6.3: Variation of pollutant concentration $c(z,t)$ in time domain and spatial domain, achieved by the learned feed flow rate $\bar{u} = 0.1232\ mh^{-1}$. At the boundary $z = 1$, $c(z,t)$ goes into the ϵ-neighborhood of desired output $y^* = 0.3gl^{-1}$ with 11 h.

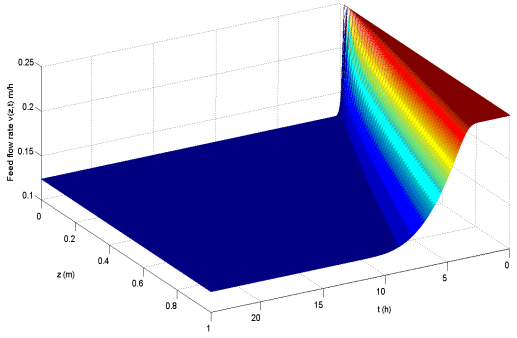


Figure 6.4: Variation of the feed flow rate $v(z,t)$ in time domain and spatial domain, by setting the boundary condition be $\bar{u} = 0.1232\ mh^{-1}$ at $z = 0$. At the boundary $z = 1$, $v(z,t)$ goes into the ϵ-neighborhood of its steady state $\bar{v} = \bar{u}$ within 10 h.

which is actually an upper bound of the input-output gradient. In (6.17), this bound is the worst case estimation for general nonlinear systems. Therefore, an iterative learning gain outside the allowable interval could also induce a learning convergence although it is not theoretically guaranteed. For instance, choose $\rho = 1.8$ and the corresponding control performance can be seen from Fig. 6.5, therein the steady output achieves the desired set-point within 10 iterations.

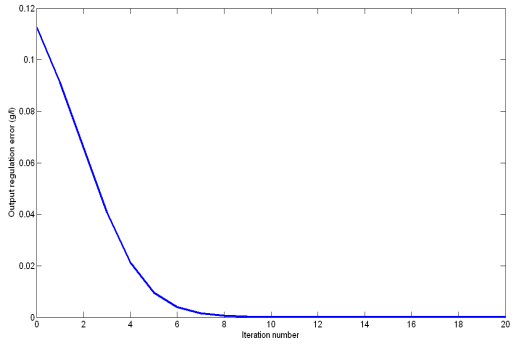


Figure 6.5: Output regulation error profile by using the proposed IBLC controller with $\rho = 1.8$.

6.5 Conclusion

We proved that a class of nonlinear PDE processes can be regulated by the IBLC control. Focusing on the steady-state performance of the system, the IBLC algorithm is designed with rigorous analysis on convergence and learning rate. The advantages of the proposed controller are its simple structure, strict convergence ensurance, and capability of dealing with input saturations easily. We show that the proposed algorithm is also applicable to more sophisticated PDE systems under certain rational assumptions.

Our last phase in this thesis is to apply the iterative learning approach to the PID parameters' tuning problem, as shown in the following chapter. The repetitiveness of control processes is assumed as well.

Chapter 7

Optimal Tuning of PID Parameters Using Iterative Learning Approach

7.1 Introduction

Among all the known controllers, the proportional-integral-derivative (PID) controllers are always the first choice for industrial control processes owing to the simple structure, robust performance, and balanced control functionality under a wide range of operating conditions. However, the exact workings and mathematics behind PID methods vary with different industrial users. Tuning PID parameters (gains) remains a challenging issue and directly determines the effectiveness of PID controllers [11, 33, 62].

To address the PID design issue, much effort has been invested in developing systematic auto-tuning methods. These methods can be divided into three categories, where the classification is based on the availability of a plant model and model type, (i) non-parametric model methods; (ii) parametric model methods; (iii) model free methods.

The non-parametric model methods use partial modelling information, usually including the steady state model and critical frequency points. These methods are more

suitable for closed-loop tuning and applied without the need for extensive priori plant information [62]. Relay feedback tuning method [12, 76, 82, 116, 124] is a representative method of the first category.

The parametric model methods require a linear model of the plant – either transfer function matrix or state space model. To obtain such a model, standard off-line or on-line identification methods are often employed to acquire the model data. Thus parametric model methods are more suitable for off-line PID tuning [10]. When the plant model is known with a parametric structure, optimal design methods can be applied [55, 70]. As for PID parameter tuning, it can be formulated as the minimization of an objective function with possible design specifications such as the nominal performance, minimum input energy, robust stability, operational constraints, etc.

In model free methods, no model or any particular points of the plant are identified. Three typical tuning methods are unfalsified control [100], iterative feedback tuning [75] and extreme seeking [66]. In [100], input-output data is used to determine whether a set of PID parameters meets performance specifications and these PID parameters are updated by an adaptive law based on whether or not the controller falsifies a given criterion. In [75], the PID controller is updated through minimizing an objective function that evaluates the closed-loop performance and estimating the system gradient. In [66], adaptive updating is conducted to tune PID parameters such that the output of the cost function reaches a local minimum or local maximum.

In practice, when the plant model is partially unknown, it would be difficult to compute PID parameters even if the relationship between transient specifications and PID parameters can be derived. In many of existing PID tuning methods, whether model free or model based, test signals will have to be injected into the plant in order to

find certain relevant information for setting controller parameters. This testing process may however be unacceptable in many real-time control tasks. On the other hand, many control tasks are carried out repeatedly, such as in batch processors. The first objective of this work is to explore the possibility of fully utilizing the task repetitiveness property, consequently provide a learning approach to improve PID controllers through iteratively tuning parameters when the transient behavior is of the main concern.

In most learning methods including neural learning and iterative learning, the process Jacobian or gradient plays the key role by providing the greatest descending direction for the learning mechanism to update inputs. The convergence property of these learning methods is solely dependent on the availability of the current information on the gradient. The gradient between the transient control specifications and PID parameters, however, may not be available if the plant model is unknown or partially unknown. Further, the gradient is a function of PID parameters, thus the magnitude and even the sign may vary. The most difficult scenario is when we do not know the sign changes *a priori*. In such circumstances, traditional learning methods cannot achieve learning convergence. The second objective of this work is to extend the iterative learning approach to deal with the unknown gradient problem for PID parameter tuning.

It should be noted that in many industrial control problems such as in process industry, the plant is stable in a wide operation range under closed-loop PID, and the major concern for a PID tuning is the transient behaviors either in the time domain, such as peak overshoot, rise time, settling time, or in the frequency domain such as bandwidth, damping ratio and undamped natural frequency. From the control engineering point of view, it is one of the most challenges to directly address the transient performance, in comparison with the stability issues, by means of tuning control parameters. Even for

a lower order linear time invariant plant under PID, the transient performance indices such as overshoot could be highly nonlinear in PID parameters and an analytical inverse mapping from overshoot to PID parameters may not exist. In other words, from the control specification on overshoot we are unable to decide the PID parameters analytically. The third objective of this work is to link these transient specifications with PID parameters and give a systematic tuning method.

Another issue is concerned with the redundancy in PID parameter tuning when only one or two transient specifications are required, for instance when only overshoot is specified, or only the integrated absolute error is to be minimized. In order to fully utilize the extra degrees of freedom of the controller, the most common approach is to introduce an objective function and optimize the PID parameters accordingly. This traditional approach is however not directly applicable because of the unknown plant model, and in particular the unknown varying gradient. A solution to this problem is still iterative learning. An objective function, which is accessible, is chosen as the first step for PID parameter optimization. Since the goal is to minimize the objective function, the control inputs will be updated along the greatest descending direction, namely the gradient, of the objective function. In other words, the PID parameters are chosen to directly reduce the objective function, and the objective function is treated as the plant output and used to update the PID parameters. When the gradient is varying and unknown, extra learning trials will be conducted to search the best descending direction.

The chapter is organized as follows. Section 7.2 gives the formulation of PID auto-tuning problem. Section 7.3 introduces the iterative learning tuning approach. Section 7.4 shows the comparative studies on benchmark examples. Furthermore, Section 7.5 addresses the real-time implementation on a laboratory pilot plant. Section 7.6 concludes

the work.

7.2 Formulation of PID Auto-tuning Problem

7.2.1 PID auto-tuning

In the following, we consider a fundamental PID controller in continuous or discrete-time

$$\begin{aligned} C(s) &= k_p + k_i\frac{1}{s} + k_d s \\ C(z) &= k_p + k_i\frac{T_s z}{z-1} + k_d\frac{z-1}{T_s z} \end{aligned}$$

where k_p is the proportional gain, k_i the integral gain, k_d is the derivative gain, s is the Laplace operator, z is the Z operator, T_s is the sampling period. Denote $\mathbf{k} = [k_p,\ k_i,\ k_d]^T$.

The aim of PID tuning is to find appropriate values for PID parameters such that the closed-loop response can be significantly improved when comparing with the open-loop response. Since the PID control performance is determined by PID parameters $\mathbf{k}$, by choosing a set of performance indices $\mathbf{x}$, for instance overshoot and settling time, there exists a unique relationship or mapping $\mathbf{f}$ between $\mathbf{x}$ and $\mathbf{k}$

$$\mathbf{x} = \mathbf{f}(\mathbf{k}).$$

The PID auto-tuning problem can be mathematically formulated as to look for a set of $\mathbf{k}$ such that $\mathbf{x}$ meet the control requirements specified by $\mathbf{x}_d$. If the inverse mapping is available, we have $\mathbf{k} = \mathbf{f}^{-1}(\mathbf{x}_d)$. The mapping $\mathbf{f}$, however, is a vector valued function of $\mathbf{x}$, $\mathbf{k}$ and the plant, and is in general a highly nonlinear mapping. Thus its inverse mapping in general is either not analytic solvable or not uniquely existing. Above all, the most difficult problem in PID auto-tuning is the lack of plant model, hence the mapping $\mathbf{f}$ is unknown or partially unknown.

The importance of PID and challenge in PID auto-tuning attracted numerous researchers and come up with various auto-tuning or detuning methods, each has unique advantages and limitations. In this work we propose a new auto-tuning method using iterative learning and optimization, which complements existing PID tuning methods.

7.2.2 Performance requirements and objective functions

An objective function, or cost function, quantifies the effectiveness of a given controller in terms of the closed-loop response, either in time domain or frequency domain. A widely used objective function in PID auto-tuning is the integrated square error (ISE) function

$$J(\mathbf{k}) = \frac{1}{T - t_0} \int_{t_0}^{T} e^2(t, \mathbf{k}) dt, \tag{7.1}$$

where the error $e(t, \mathbf{k}) = r(t) - y(t, \mathbf{k})$ is the difference between the reference, $r(t)$, and the output signal of the closed-loop system, $y(t)$. T and t_0, with $0 \leq t_0 < T < \infty$, are two design parameters. In several auto-tuning methods such as IFT and ES, t_0 is set approximately at the time when the step response of the closed-loop system reaches the first peak. Hence the cost function effectively places zero weighting on the initial transient portion of the response and the controller is tuned to minimize the error beyond the peak time. Similar objective functions, such as integrated absolute error, integrated time weighted absolute error, integrated time weighted square error,

$$\frac{1}{T - t_0} \int_{t_0}^{T} |e| dt, \quad \frac{1}{T - t_0} \int_{t_0}^{T} t|e| dt, \quad \frac{1}{T - t_0} \int_{t_0}^{T} te^2 dt,$$

have also been widely used in the process of PID parameter tuning.

In many control applications, however, the transient performance, such as overshoot M_p, settling time t_s, rise time t_r, could be of the main concern. An objective function

that can capture the transient response directly, is highly desirable. For this purpose, we propose a quadratic function

$$J = (\mathbf{x}_d - \mathbf{x})^T Q(\mathbf{x}_d - \mathbf{x}) + \mathbf{k}^T R\mathbf{k}, \tag{7.2}$$

where $\mathbf{x} = [100M_p, t_s, t_r]^T$, $\mathbf{x}_d$ is the desired $\mathbf{x}$, Q and R are two constant non-negative weighting matrices. The selection of Q and R matrices will yield effects on the optimization results, that is, produce different closed-loop responses. A larger Q highlights more on the transient performance, whereas a larger R gives more control penalty.

7.2.3 A second order example

Consider a second order plant under a unity feedback with a PD controller. The PD controller and plant are respectively

$$C(s) = k_p + k_d s, \qquad G(s) = \frac{k}{s^2 + as + b}, \tag{7.3}$$

where k_p, k_d, k, a, b are all non-negative constants. The closed-loop system is stable if $a + d > 0$ and $b + p > 0$, where $p = kk_p$ and $d = kk_d$.

The nonlinear mapping $\mathbf{f}$ between the closed-loop transient response $\mathbf{x} = [M_p,\ t_s]$ and the PD parameters $\mathbf{k} = [k_p, k_d]$ is derived in Appendix A, and shown in Fig. 7.1 and Fig. 7.2, where $a = 0.1$, $b = 0$, and $k = 1$. It can be seen that the mapping $\mathbf{f}$ is nonlinear or even discontinuous.

The nonlinear mapping $\mathbf{f}$ for discrete-time control system can also be derived but omitted here due to the complexity. Discretizing the plant in (7.3) with a sampling time $T_s = 0.1$ s, the nonlinear mapping between M_p and (k_p, k_d) is shown in Fig. 7.3. It can be seen that there exist local minima in the surface, and the gradient may vary and take either positive or negative values.

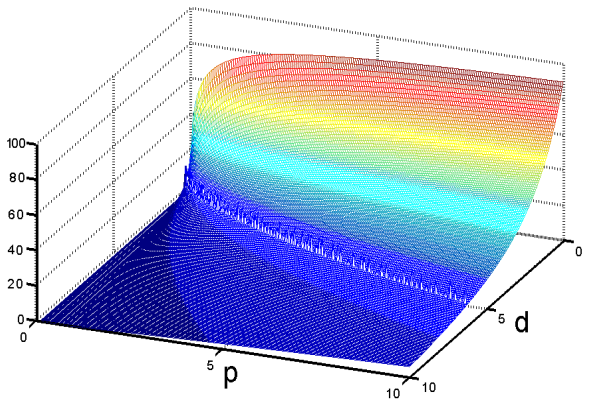


Figure 7.1: The nonlinear mapping between the peak overshoot $100M_p$ and PD gains (k_p, k_d) in continuous-time.

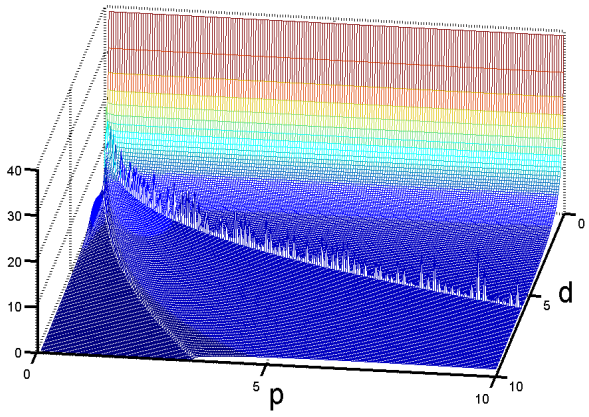


Figure 7.2: The nonlinear mapping between the settling time t_s and PD gains (k_p, k_d) in continuous-time.

On the other hand, it can also been seen from those figures that the transient responses may vary drastically while the control parameters only vary slightly. This indicates the importance for PID parameter auto-tuning and the necessity for finding an effective tuning method.

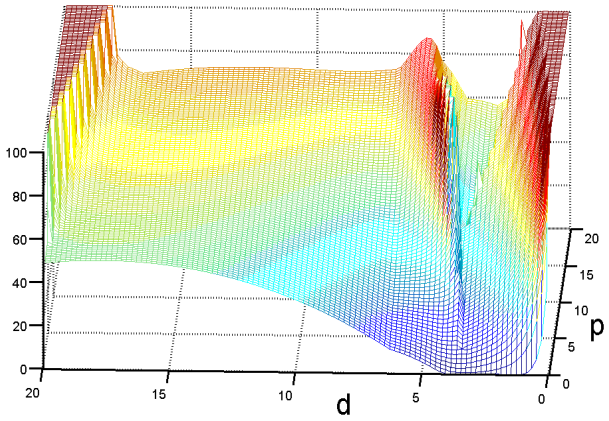


Figure 7.3: The nonlinear mapping between the peak overshoot $100M_p$ and PD gains (k_p, k_d) in discrete-time. An upper bound of 100 is applied to crop the vertical values.

7.3 Iterative Learning Approach

Iterative learning is adopted to provide a solution to the PID auto-tuning. The iterative learning offers a desirable feature that it can guarantee the learning convergence even if the plant model is partially unknown.

7.3.1 Principal idea of iterative learning

The concept of iterative learning was first introduced in control to deal with a repeated control task without requiring the perfect knowledge such as the plant model or parameters [8]. It learns to generate a control action directly instead of doing a model identification. The iterative learning mechanism updates the present control action using information obtained from previous control actions and previous error signals.

Let us first give the basic formulation of iterative learning in PID auto-tuning. From preceding discussions, the PID auto-tuning problem can be described by the mapping

$$\mathbf{x} = \mathbf{f}(\mathbf{k})$$

where $\mathbf{k} \in \Omega_k \subset \mathcal{R}^n$ and $\mathbf{x} \in \Omega_x \subset \mathcal{R}^m$, where n and m are integer numbers. The

learning objective is to find a suitable set $\mathbf{k}$ such that the transient response $\mathbf{x}$ can reach a given region around the control specifications $\mathbf{x}_d$.

The principal idea of iterative learning is to construct a contractive mapping $\mathcal{A}$

$$\mathbf{x}_d - \mathbf{x}_{i+1} = A(\mathbf{x}_d - \mathbf{x}_i),$$

where the norm of A is strictly less than 1, and the subscript i indicates that the quantity is in the ith iteration or learning trial. To achieve this contractive mapping, a simple iterative learning law is

$$\mathbf{k}_{i+1} = \mathbf{k}_i + \Gamma_i(\mathbf{x}_d - \mathbf{x}_i) \tag{7.4}$$

where $\Gamma_i \in \mathcal{R}^{n\times m}$ is a learning gain matrix. It can be seen that the learning law (7.4) generates a set of updated parameters from the previously tuned parameters, $\mathbf{k}_i$, and previous performance deviations $\mathbf{x}_d - \mathbf{x}_i$. The schematic of the iterative learning mechanism for PID auto-tuning is shown in Fig. 7.4.

When $n = m$, define the process gradient

$$F(\mathbf{k}) = \frac{\partial \mathbf{f}(\mathbf{k})}{\partial \mathbf{k}},$$

we can derive the condition for the contractive mapping $\mathcal{A}$

$$\begin{aligned}
\mathbf{x}_d - \mathbf{x}_{i+1} &= \mathbf{x}_d - \mathbf{x}_i - (\mathbf{x}_{i+1} - \mathbf{x}_i) \\
&= \mathbf{x}_d - \mathbf{x}_i - \frac{\partial \mathbf{f}(\mathbf{k}_i^*)}{\partial \mathbf{k}}(\mathbf{k}_{i+1} - \mathbf{k}_i) \\
&= [I - F(\mathbf{k}_i^*)\Gamma_i](\mathbf{x}_d - \mathbf{x}_i) \qquad (7.5)
\end{aligned}$$

where $\mathbf{k}_i^* \in [\min\{\mathbf{k}_i, \mathbf{k}_{i+1}\},\ \max\{\mathbf{k}_i, \mathbf{k}_{i+1}\}] \subset \Omega_k$. Therefore we have a contractive mapping $\mathcal{A}$

$$\mathbf{x}_d - \mathbf{x}_{i+1} = A(\mathbf{x}_d - \mathbf{x}_i)$$

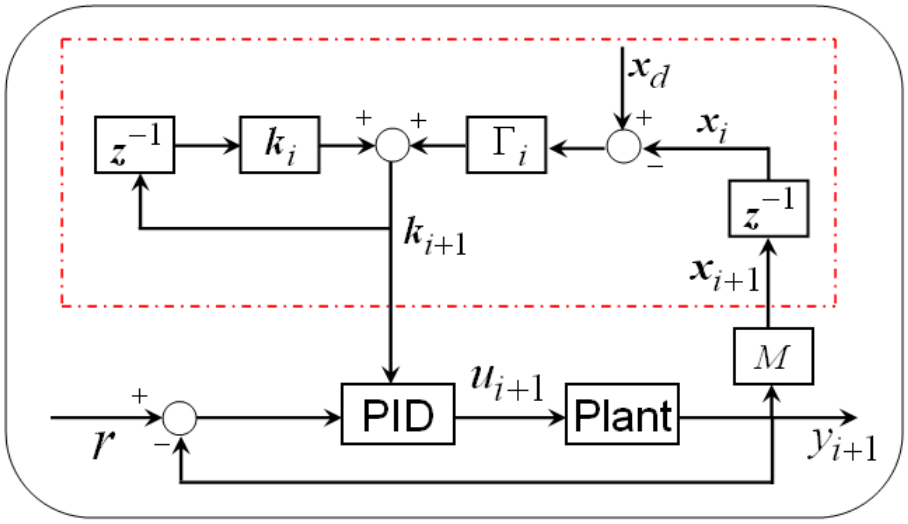


Figure 7.4: The schematic block diagram of the iterative learning mechanism and PID control loop. The parameter correction is generated by the performance deviations $\mathbf{x}_d - \mathbf{x}_i$ multiplied by a learning gain Γ_i. The operator $\mathbf{z}^{-1}$ denotes one iteration delay. The new PID parameters $\mathbf{k}_{i+1}$ consists of the previous $\mathbf{k}_i$ and the correction term, analogous to a discrete-time integrator. The iterative learning tuning mechanism is shown by the block enclosed by the dashed line. r is the desired output and the block M is a feature extraction mechanism that records the required transient quantities such as overshoot from the output response y_{i+1}.

as far as the magnitude

$$|A| = |I - F(\mathbf{k}_i^*)\Gamma_i| \leq \rho < 1. \tag{7.6}$$

When $n > m$, there exists an infinite number of solutions because of redundancy in control parameters. With the extra degrees of freedom in PID, optimality can be exploited, for instance the shortest settling time, minimum peak overshoot, minimum values of control parameters, etc. A suitable objective function to be minimized could be a non-negative function $J(\mathbf{x}, \mathbf{k}) \geq 0$, where J is accessible, such as the quadratic one (7.2). The minimization is in fact a searching task

$$\min_{\mathbf{k} \in \Omega_k} J(\mathbf{x}, \mathbf{k})$$

where $\mathbf{x} = \mathbf{f}(\mathbf{k})$. The optimal parameters $\mathbf{k}_{opt}$ can be obtained by differentiating J and computing

$$\frac{dJ}{d\mathbf{k}} = \frac{\partial J}{\partial \mathbf{x}}\frac{d\mathbf{x}}{d\mathbf{k}} + \frac{\partial J}{\partial \mathbf{k}} = \frac{\partial J}{\partial \mathbf{x}}F + \frac{\partial J}{\partial \mathbf{k}} = 0. \tag{7.7}$$

In most cases $\mathbf{k}_{opt}$ can only be found numerically due to the highly nonlinear relationship between $\mathbf{k}$ and quantities J, $\mathbf{x}$. A major limitation of optimization methods is the demand for the complete plant model knowledge or the mapping $\mathbf{f}$. On the contrary, iterative learning only requires the bounding knowledge of the process gradient. The principal idea of iterative learning can be extended to solving the optimization problem under the assumption that all gradient components with respect to $\mathbf{k}$ have known limiting bounds. Since the learning objective now is to directly reduce the value of J, the objective function J can be regarded as the process output to be minimized. The new iterative learning tuning law is

$$\mathbf{k}_{i+1} = \mathbf{k}_i - \boldsymbol{\gamma}_i J(\mathbf{k}_i), \tag{7.8}$$

where $\boldsymbol{\gamma}_i = [\gamma_{1,i}, \cdots, \gamma_{n,i}]^T$ and $J(\mathbf{k}_i)$ denotes $J(\mathbf{k}_i, \mathbf{f}(\mathbf{k}_i))$. To show the contractive mapping, note that

$$\begin{aligned} J(\mathbf{k}_{i+1}) &= J(\mathbf{k}_i) + [J(\mathbf{k}_{i+1}) - J(\mathbf{k}_i)] \\ &= J(\mathbf{k}_i) + \left(\frac{dJ(\mathbf{k}_i^*)}{d\mathbf{k}}\right)^T (\mathbf{k}_{i+1} - \mathbf{k}_i) \end{aligned}$$

where $\mathbf{k}_i^* \in \Omega_k$ is in a region specified by $\mathbf{k}_i$ and $\mathbf{k}_{i+1}$. By substitution of the ILT law (7.8), we have

$$J(\mathbf{k}_{i+1}) = \left[1 - \left(\frac{dJ(\mathbf{k}_i^*)}{d\mathbf{k}}\right)^T \boldsymbol{\gamma}_i\right] J(\mathbf{k}_i). \tag{7.9}$$

The convergence property is determined by the learning gains $\boldsymbol{\gamma}_i$ and the gradient $dJ/d\mathbf{k}$.

7.3.2 Learning gain design based on gradient information

To guarantee the contractive mapping (7.9), the magnitude relationship must satisfy $|1 - D_i^T \boldsymbol{\gamma}_i| \leq \rho < 1$, where

$$D_i = \left(\frac{dJ(\mathbf{k}_i^*)}{d\mathbf{k}} \right)$$

is the gradient of the objective function. The selection of learning gain $\boldsymbol{\gamma}_i$ is highly related to the prior knowledge on the gradient D_i. Consider three scenarios and assume $D_i = (D_{1,i}, D_{2,i}, D_{3,i})$, that is, a PID controller is used.

When D_i is known *a priori*, we can choose $\gamma_{j,i} = D_{j,i}^{-1}/3$. Such a selection produces the fastest learning convergence speed, that is, convergence in one iteration because $\|1 - D_i^T \boldsymbol{\gamma}_i\| = 0$.

When the bounding knowledge and the sign information of $D_{j,i}$ are available, the learning convergence can also be guaranteed. For instance assume $0 < \alpha_j \leq D_{j,i} \leq \beta_j < \infty$ for $\mathbf{k}_i^* \in \Omega_k$, where α_j and β_j are respectively lower and upper bounds of the gradient components $D_{j,i}$. In such circumstances, choosing $\gamma_{j,i} = 1/3\beta_j$, the upper bound of the magnitude is $\rho = 1 - \alpha_1/3\beta_1 - \alpha_2/3\beta_2 - \alpha_3/3\beta_3 < 1$.

The most difficult scenario is when bounding functions or signs of $D_{j,i}$ are unknown. In order to derive the iterative learning convergence, it can be seen from (7.9) that we do not need the exact knowledge about the mapping $\mathbf{f}$. It is adequate to know the bounding knowledge (amplitude and sign) of the gradient $D(\mathbf{k}_i^*)$ or $F(\mathbf{k}_i^*)$. Although an invariant gradient is assumed for most iterative learning problems, the elements in $D(\mathbf{k}_i^*)$ may change sign and take either positive or negative values. Without knowing the exact knowledge of the mapping $\mathbf{f}$, we may not be able to predict the varying signs in the gradient. Now, in the iterative learning law (7.8) the learning gains will have to change signs according to the gradient. Here the question is how to change the signs of

the learning gains when we do not know the signs of the gradient components, i.e. how to search the direction of the gradient, which obviously can only be done in a model free manner if $\mathbf{f}$ is unknown.

A solution to the problem with unknown gradient is to conduct extra learning trials to determine the direction of gradient or the signs of the learning gains *directly*

$$\boldsymbol{\gamma}_i = [\pm\gamma_1, \cdots, \pm\gamma_n]^T, \tag{7.10}$$

where γ_i are positive constants. From the derivation (7.5), when learning gains are chosen appropriately, $|A| < 1$ and the learning error reduces. On the other hand, if learning gains are chosen inappropriately, then $|A| > 1$ and the error increases after this learning trial. Therefore, several learning trials are adequate for ILT mechanism to determine the correct signs of the learning gains.

In general, when there are two gradient components $(D_{1,i}, D_{2,i})$, there are 4 sets of signs $\{1, 1\}$, $\{1, -1\}$, $\{-1, 1\}$, and $\{-1, -1\}$, corresponding to all possibles signs of the gradient $(D_{1,i}, D_{2,i})$. In such circumstances, at most 4 learning trials are sufficient to find the greatest descending among the four control directions, as shown in Fig. 7.5.

Similarly, if there are three free tuning parameters, there will be 8 sets of signs in the gradient $(D_{1,i}, D_{2,i}, D_{3,i})$, as shown in Fig. 7.6. In general, if there are n control tuning parameters, there will be n gradient components $D_{j,i}$. Since each $\gamma_{j,i}$ takes either positive or negative sign, there will be 2^n combinations and at most 2^n learning trials are required.

In addition to the estimation of the gradient direction or learning direction, the magnitudes of learning gains $\boldsymbol{\gamma}_i$ should also be adjusted to satisfy the learning convergence condition $|1 - D_i^T\boldsymbol{\gamma}_i| \leq \rho < 1$. Since the gradient D_i is a function of PID parameters $\mathbf{k}_i$, the magnitude of D_i varies at different iterations. When the magnitudes of the gradient

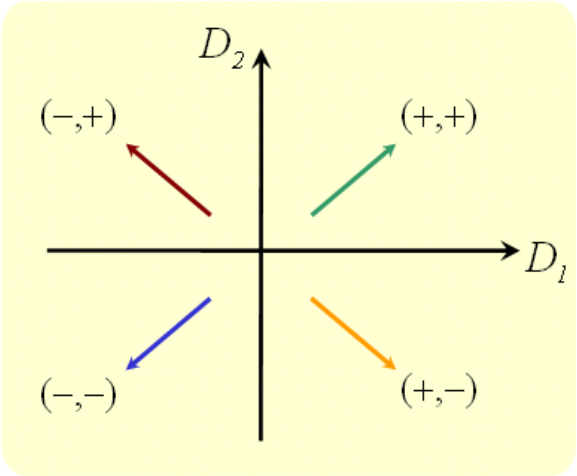


Figure 7.5: There are four pairs of signs for the gradient (D_1, D_2) as indicated by the arrows. Hence there are four possible updating directions, in which one pair gives the fastest descending direction.

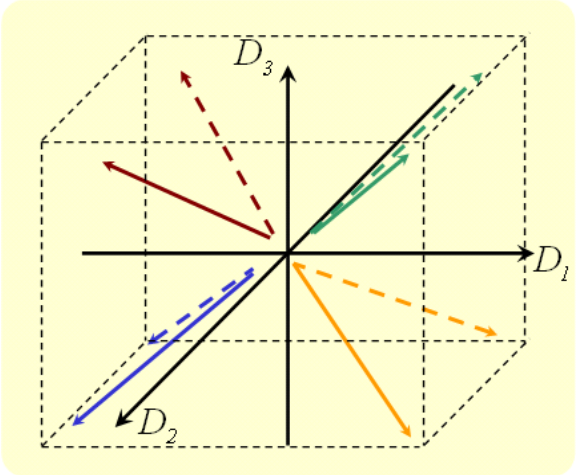


Figure 7.6: There are three gradient components D_1, D_2 and D_3 with respect to three control parameters. Consequently there are 8 possible tuning directions and at most 8 learning trials are required to find the correct updating direction.

is unknown, extra learning trials will be needed to search for suitable magnitudes of learning gains.

In this work, we adopt a self-adaption rule [102] which scales learning gains up and down by a factor $\zeta = 1.839$, that is, each component $\gamma_{j,i}$ will be adjusted to $\gamma_{j,i}\zeta$ and $\gamma_{j,i}/\zeta$. Extra learning trials will be performed with the scaled learning gains. It is

reported that the choice of such a scaling factor ζ will lead to a linear convergence rate [102, 103] .

To facilitate searching and reduce the number of trials, we can also estimate gradient components $D_{j,i}$ numerically using the values of the objective function and PID parameters obtained from previous two iterations

$$\hat{D}_{j,i} = \frac{J(\mathbf{k}_{i-1}) - J(\mathbf{k}_{i-2})}{k_{j,i-1} - k_{j,i-2}}. \tag{7.11}$$

The learning gain can be revised accordingly as $\gamma_{j,i} = \lambda_j \hat{D}_{j,i}^{-1}$ where λ_j is a constant gain in the interval of $(0, 1]$. This method is in essence the Secant method along the iteration axis, which can effectively expedite the learning speed [148].

Since gradient direction is a critical issue in searching, and the approximation (7.11) may not always guarantee a correct sign, we can use the estimation result in (7.11) partially by retaining the magnitude estimation, while still searching the correct control direction

$$\gamma_{j,i} = \pm\lambda_j \left| \frac{k_{j,i-1} - k_{j,i-2}}{J(\mathbf{k}_{i-1}) - J(\mathbf{k}_{i-2})} \right|. \tag{7.12}$$

7.3.3 Iterative searching methods

Three iterative searching methods are considered for ILT in this work with verifications and comparisons. They are M_0 – an exhaustive searching method in directions and magnitude, M_1 – an exhaustive searching method in directions, M_2 – a lazy searching method.

M_0 does exhaustive searching in all 2^n directions and exhaustive searching in all 2^n magnitudes using self-adaptation with the factor ζ. Then PID parameters will be updated using the set that generates the best closed-loop response or yields the biggest drop of J in that trial. With the best tuned PID parameters as the initial setting, the

ILT mechanism enters another run of exhaustive searching for the best response and best PID parameters. The searching process repeats until the stopping criterion is met.

This is the worst case searching where neither the gradient directions nor the gradient magnitudes are available. For each run of searching the greatest descending direction, 4^n trials are performed. For PID tuning where $n = 3$, 64 trials are needed for one run of searching. Clearly, this searching method is not efficient.

In M_1 the entire searching and updating process is similar to M_0 except that the magnitudes of learning gains are determined using the formula (7.12). Hence it reduces the total number of trials from 4^n in M_0 to 2^n for each run of searching. For PID tuning where $n = 3$, only 8 trials are needed.

M_2 does exhaustive searching in directions in the first run of searching, and the magnitudes of learning gains are determined using the formula (7.12). Then the greatest descending direction will be used for subsequent runs of searching, with the assumption that the mapping $\mathbf{f}$ is in general smooth and drastic variations in gradient rarely occur. The exhaustive searching in directions will be activated again when the stopping criterion is met. The searching process will permanently stop if the stop criterion is still met after exhaustive searching in all directions.

The initial magnitudes of learning gains can be set as

$$[\gamma_{1,0}, \gamma_{2,0}, \gamma_{3,0}] = \frac{\gamma_0}{J_0}[k_{p,0}, k_{i,0}, k_{d,0}], \tag{7.13}$$

where γ_0 is a positive constant, and chosen to be 0.1 in this work. $k_{p,0}$, $k_{i,0}$, $k_{d,0}$ are initial values of PID parameters determined using any existing PID auto-tuning methods. J_0 is calculated with $k_{p,0}$, $k_{i,0}$, $k_{d,0}$ and the corresponding closed-loop response.

7.4 Comparative Studies on Benchmark Examples

In this section we conduct comprehensive tests on 8 benchmark plant models. Four plant models G_1 to G_4 were used in [66] to compare and validate several iterative tuning methods

$$\begin{aligned} G_1(s) &= \frac{1}{1+20s}e^{-5s}, \\ G_2(s) &= \frac{1}{1+20s}e^{-20s}, \\ G_3(s) &= \frac{1}{(1+10s)^8}, \\ G_4(s) &= \frac{1-5s}{(1+10s)(1+20s)}, \end{aligned}$$

where G_1 and G_2 have relatively large normalized time delay (NTD), G_3 has high-order repeated poles, and G_4 is nonminimum phase. The other four plant models G_5 to G_8 were used in [49, 126] to validate their auto-tuning methods

$$\begin{aligned} G_5(s) &= \frac{1}{(s+1)(s+5)^2}e^{-0.5s}, \\ G_6(s) &= \frac{1}{(25s^2+5s+1)(5s+1)}e^{-s}, \\ G_7(s) &= \frac{1}{(s^2+2s+3)(s+3)}e^{-0.3s}, \\ G_8(s) &= \frac{1}{(s^2+s+1)(s+2)^2}e^{-0.1s}, \end{aligned}$$

where G_5 is a high-order plant with medium NTD, G_6 is a high-order and moderately oscillatory plant with short NTD, G_7 is a high-order and heavily oscillatory plant with short NTD, and G_8 has both oscillatory and repeated poles.

To make fair comparisons, we choose initial learning gains with form (7.13) for PID parameters in all case studies. In all searching results, we use N to denote the total number of trials.

7.4.1 Comparisons between objective functions

Objective function ISE (7.1) has been widely investigated and adopted in PID tuning. The quadratic objective function (7.2), on the other hand, has more weights to prioritize transient performance requirements. When control requirements are directly concerned with transient response such as M_p or t_s, we can only use the quadratic objective function (7.2). To show the effects of different objective functions and weight selections, we use plants G_1 – G_4. To make fair comparisons, the learning process starts from the same initial setting for PID gains generated by Ziegler-Nicholes (ZN) tuning method [10]. Exhaustive searching method M_0 is employed. The tuning results through iterative learning are summarized in Table 4.1. For simplicity only M_p and t_s are taken into consideration in (7.2). The learning process stops when the drop of an objective function between two consecutive iterations is lower than $\varepsilon = 10^{-6}$ for (7.1) and $\varepsilon = 0.01$ for (7.2). In ISE, the parameters are $T = 100, 300, 500, 200$ s and $t_0 = 10, 50, 140, 30$ s respectively [66].

Usually settling time is much greater than overshoot, thus $100M_p$ is used. When plants have much bigger settling time, we can choose t_s instead of $(t_s)^2$ in the objective function, as shown in cases of G_2 and G_3. By scaling down t_s in objective functions, overshoot decreases as it is weighted more. Meanwhile t_s increases as it is weighted less.

Comparing ISE and quadratic objective function, we can see that latter offers more choices.

7.4.2 Comparisons between ILT and existing iterative tuning methods

Now we compare iterative learning based tuning with other iterative tuning methods such as extremum seeking (ES) tuning and iterative feedback tuning (IFT). ZN tuning

Plant	J	PID Controller	$100M_p$	t_s	N
G_1	ISE	$3.59 + 0.13s^{-1} + 7.54s$	4.48	21.35	768
	$(100M_p)^2 + 0.5t_s^2$	$3.38 + 0.13s^{-1} + 7.05s$	1.99	12.17	1024
	$(100M_p)^2 + 0.1t_s^2$	$3.25 + 0.12s^{-1} + 6.30s$	0.63	12.83	832
	$(100M_p)^2 + 0.01t_s^2$	$3.71 + 0.11s^{-1} + 9.11s$	0.53	22.24	512
G_2	ISE	$0.93 + 0.031s^{-1} + 5.67s$	0.71	50.67	512
	$(100M_p)^2 + t_s$	$0.99 + 0.032s^{-1} + 6.87s$	1.06	47.99	1600
	$(100M_p)^2 + 0.2t_s$	$1.05 + 0.028s^{-1} + 9.79s$	0.29	82.74	512
	$(100M_p)^2 + 0.1t_s$	$1.03 + 0.029s^{-1} + 9.18s$	0.20	83.61	640
G_3	ISE	$0.64 + 0.012s^{-1} + 11.3s$	0.49	137.13	1024
	$(100M_p)^2 + t_s$	$0.76 + 0.013s^{-1} + 16.65s$	1.93	120.56	576
	$(100M_p)^2 + 0.2t_s$	$0.85 + 0.014s^{-1} + 25.77s$	0.66	212.04	192
	$(100M_p)^2 + 0.1t_s$	$0.83 + 0.014s^{-1} + 24.91s$	0.62	212.76	192
G_4	ISE	$5.01 + 0.092s^{-1} + 25.59s$	3.05	25.2	1216
	$(100M_p)^2 + 0.25t_s^2$	$4.31 + 0.075s^{-1} + 22.19s$	1.81	18.63	512
	$(100M_p)^2 + 0.1t_s^2$	$3.89 + 0.071s^{-1} + 22.28s$	1.70	20.56	384
	$(100M_p)^2 + 0.01t_s^2$	$4.51 + 0.075s^{-1} + 23.96s$	0.06	19.27	1216

Table 7.1: Control performances of $G_1 - G_4$ using the proposed ILT method.

method is also included for comparison. G_1 – G_4 and the same ISE [66] for each model are used. The PID controller parameters given by ZN are used as a starting point for ILT tuning. Exhaustive searching method M_0 is employed in ILT.

The results verify that ILT, ES and IFT give very similar responses which are much superior than that of ZN tuning method. Fig. 7.7 shows the ILT results for G_1, (a) shows the decreasing objective function J, (b) shows decreasing performance indices M_p and t_s, (c) shows the variations of the PID parameters, and (d) compares step responses with four tuning methods. Fig. 7.8 shows the searching results of the gradient directions and the variations of the learning gains through self-adaptation. It can be seen that the gradients undergo changes in signs, hence it is in general a difficult and challenging optimization problem.

Table 4.2 summarizes the comparative results with all four plants G_1 – G_4 when 4 tuning methods were applied. The iteration numbers when applying ILT for G_1 – G_4 are 786, 512, 1024 and 1216 respectively.

Although ILT shares similar performance as ES and IFT, it is worth to highlight some important factors in the tuning process. In ES tuning, there are more than 10 design parameters to be set properly. From [66], design parameters take rather specific values. In IFT, the initial values of the PID parameters must be chosen in such a way as to give an initial response that is very slow and with no overshoot. Further, in IFT the transient performance is purposely excluded from the objective function by choosing a sufficiently large t_0. On the contrary, in ILT only initial learning gains need to be preset, and we choose all three learning gains with a uniform value $\gamma_0 = 0.1$ in (7.13) for the ease of ILT design. In fact, by choosing initial learning gains with different values, we can achieve much better responses than those in Table 4.2. Needless to mention that

Plant	Method	PID Controller	$100M_p$	t_s
G_1	ZN	$4.06 + 0.44s^{-1} + 9.38s$	46.50	47.90
	IFT	$3.67 + 0.13s^{-1} + 7.74s$	5.38	21.38
	ES	$3.58 + 0.13s^{-1} + 7.68s$	3.31	21.35
	ILT	$3.59 + 0.13s^{-1} + 7.54s$	4.48	21.35
G_2	ZN	$1.33 + 0.043s^{-1} + 10.30s$	21.75	109.4
	IFT	$0.93 + 0.031s^{-1} + 5.64s$	0.80	50.33
	ES	$1.01 + 0.032s^{-1} + 7.23s$	1.37	76.61
	ILT	$0.93 + 0.031s^{-1} + 5.67s$	0.71	50.67
G_3	ZN	$1.10 + 0.015s^{-1} + 20.91s$	13.95	336.90
	IFT	$0.66 + 0.012s^{-1} + 12.08s$	0.98	132.05
	ES	$0.68 + 0.013s^{-1} + 13.30s$	0.96	130.41
	ILT	$0.64 + 0.012s^{-1} + 11.30s$	0.49	137.13
G_4	ZN	$3.53 + 0.21s^{-1} + 14.80s$	53.70	86.12
	IFT	$3.03 + 0.065s^{-1} + 18.42s$	0.55	28.74
	ES	$3.35 + 0.068s^{-1} + 21.40s$	0.18	29.80
	ILT	$5.01 + 0.092s^{-1} + 25.59s$	3.05	25.21

Table 7.2: Control performances of $G_1 - G_4$ using methods ZN, IFT, ES and ILT.

ILT can handle both types of objective functions (7.1) and (7.2) with the flexibility to highlight transient behaviors.

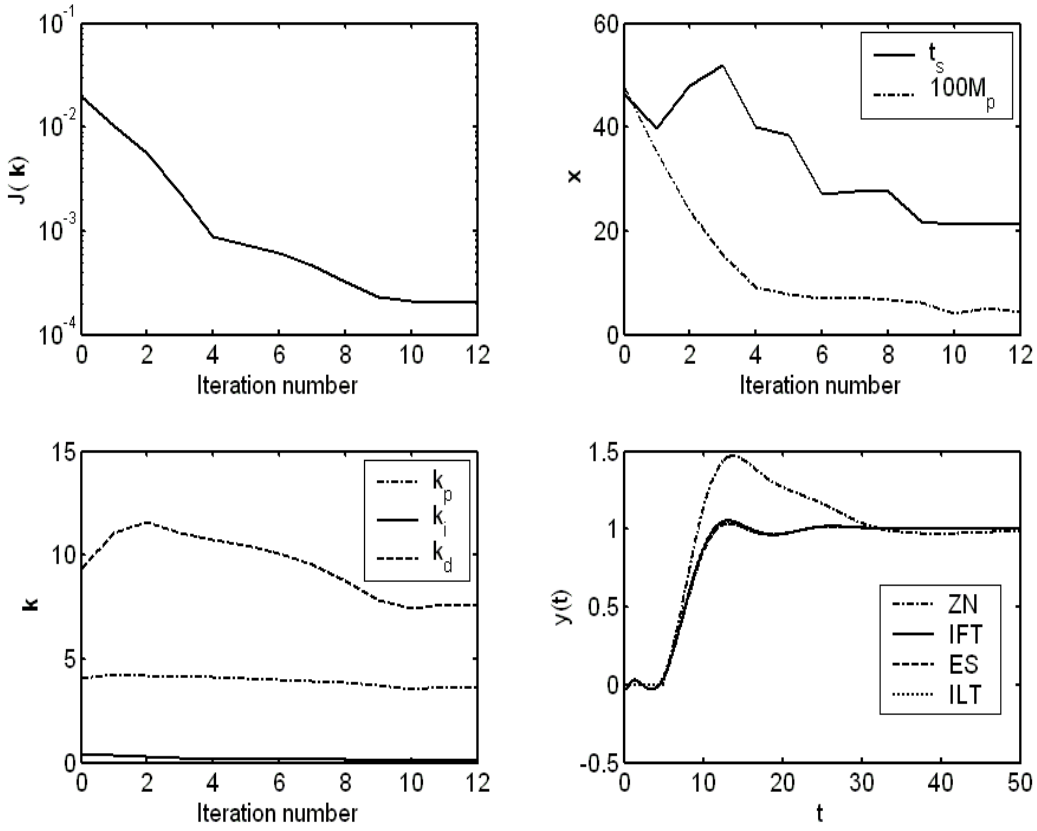


Figure 7.7: ILT performance for G_1. (a) The evolution of the objective function; (b) The evolution of overshoot and settling time; (c) The evolution of PID parameters; (d) The comparisons of step responses among ZN, IFT, ES and ILT, where IFT, ES and ILT show almost the same responses.

7.4.3 Comparisons between ILT and existing auto-tuning methods

PID auto-tuning methods [10, 49, 126] provided several effective ways to determine PID parameters. In this subsection we compare ILT with the auto-tuning method [10] based on internal model control (IMC), and the auto-tuning method [126] based on pole-placement (PPT) which shows superior performance than [49]. Comparisons are made

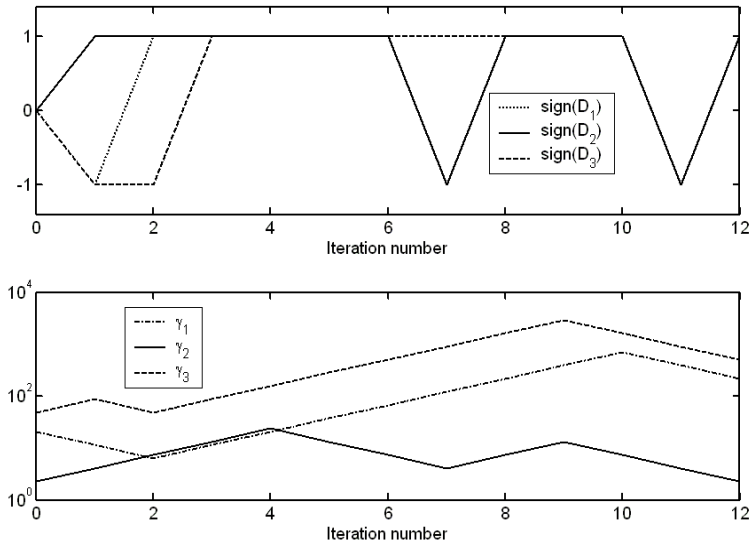


Figure 7.8: ILT searching results for G_1. (a) The evolution of the gradient directions; (b) The evolution of the magnitudes of learning gains with self-adaptation.

based on plants G_5 – G_8 which were used as benchmarks. Using ISE as the objective function and searching method M_0 in ILT, the tuning results are summarized Table 4.3, where the start points of PID parameters are adopted from the tuning results in [49]. Comparing with the results auto-tuned by the IMC method and PPT method, ILT achieves better performance after learning.

7.4.4 Comparisons between searching methods

Now we investigate the effects of searching methods M_0, M_1 and M_2. Plants G_1 – G_8 are used. The objective function to be used is

$$J = (100M_p)^2 + \frac{t_s^2}{100}. \tag{7.14}$$

Set $\lambda_j = 0.2$, $j = 1, 2, 3$ in (7.12) for M_1 and M_2. The stopping criterion for ILT is $\varepsilon = 0.01$. The control performance after learning is summarized in Tables 4.4 and 4.5.

It can be seen that M_1 and M_2 achieve similar performance, which is slightly inferior

Plant	Method	PID Controller	$100M_p$	t_s
G_5	IMC	$33.46 + 16.67s^{-1} + 8.54s$	2.09	5.71
	PPT	$27.02 + 21.14s^{-1} + 6.45s$	12.12	5.01
	ILT	$31.96 + 18.81s^{-1} + 11.01s$	1.31	2.09
G_6	IMC	$0.88 + 0.066s^{-1} + 1.20s$	3.46	74.61
	PPT	$0.44 + 0.074s^{-1} + 2.54s$	5.10	49.81
	ILT	$0.57 + 0.075s^{-1} + 2.78s$	1.17	23.91
G_7	IMC	$7.06 + 5.29s^{-1} + 1.64s$	5.15	8.37
	PPT	$3.89 + 5.39s^{-1} + 2.15s$	3.21	5.32
	ILT	$4.51 + 5.40s^{-1} + 2.53s$	0.44	2.93
G_8	IMC	$2.79 + 1.33s^{-1} + 1.19s$	3.01	14.74
	PPT	$1.50 + 1.37s^{-1} + 1.72s$	3.04	9.44
	ILT	$2.18 + 1.48s^{-1} + 2.34s$	1.18	4.61

Table 7.3: Control performances of $G_5 - G_8$ using IMC, PPT and ILT methods.

Plant	M_0			M_1			M_2		
	$100M_p$	t_s	N	$100M_p$	t_s	N	$100M_p$	t_s	N
G_1	0.53	22.24	512	0.24	22.13	160	0.23	22.23	29
G_2	1.80	78.64	256	0.73	79.93	72	1.47	79.91	43
G_3	1.81	121.29	192	1.36	209.77	48	1.81	205.35	41
G_4	0.06	19.27	1216	0	15.92	120	0	22.93	41
G_5	0.282	2.16	640	0.21	2.44	48	0	3.34	70
G_6	0.605	25.04	320	0.03	48.04	40	0.03	48.04	19
G_7	0	6.05	192	0	6.13	32	0	6.15	28
G_8	0.19	8.25	768	0	10.53	32	0	10.74	18

Table 7.4: Control performance of $G_1 - G_8$ using searching methods M_0, M_1, M_2.

Plant	M_0	M_1	M_2
G_1	$3.59 + 0.13s^{-1} + 7.54s$	$3.42 + 0.11s^{-1} + 8.77s$	$3.38 + 0.12s^{-1} + 7.80s$
G_2	$0.93 + 0.029s^{-1} + 9.43s$	$0.94 + 0.030s^{-1} + 7.28s$	$1.05 + 0.031s^{-1} + 8.98s$
G_3	$0.75 + 0.013s^{-1} + 16.26s$	$0.87 + 0.012s^{-1} + 24.73s$	$0.72 + 0.012s^{-1} + 21.42s$
G_4	$4.51 + 0.075s^{-1} + 23.96s$	$5.99 + 0.097s^{-1} + 30.70s$	$6.92 + 0.10s^{-1} + 39.18s$
G_5	$31.52 + 18.31s^{-1} + 10.76s$	$29.63 + 17.65s^{-1} + 10.19s$	$17.00 + 13.68s^{-1} + 0.54s$
G_6	$0.51 + 0.070s^{-1} + 2.14s$	$0.60 + 0.069s^{-1} + 2.17s$	$0.60 + 0.069s^{-1} + 2.17s$
G_7	$5.34 + 4.86s^{-1} + 1.28s$	$5.02 + 4.81s^{-1} + 1.31s$	$4.11 + 4.81s^{-1} + 1.31s$
G_8	$2.03 + 1.27s^{-1} + 2.71s$	$2.13 + 1.21s^{-1} + 0.94s$	$1.74 + 1.21s^{-1} + 0.94s$

Table 7.5: Final controllers for $G_1 - G_8$ by using searching methods M_0, M_1, M_2.

than M_0. However, comparing with M_0 the learning trial numbers in M_1 and M_2 have been significantly reduced.

7.4.5 ILT for sampled-data systems

A promising feature of ILT is the applicability to sampled-data or discrete-time systems. To illustrate how ILT works for digital systems, consider plant G_4 which can be discretized using sampler and zero order hold

$$G_4(z) = \frac{\left(-2.5e^{-.05T_s} + 1.5e^{-.1T_s} + 1\right) z + e^{-.15T_s} - 2.5e^{-0.1T_s} + 1.5e^{-.05T_s}}{z^2 - (e^{-0.1T_s} + e^{-0.05T_s})z + e^{-0.15T_s}},$$

where the sampling period $T_s = 0.1$ s. The digital PID controller is used. Choose again (7.14) as the objective function, and use ZN to generate the initial values for PID parameters. The closed-loop responses using ILT are summarized in Table 4.6. For comparison all three searching methods M_0, M_1 and M_2 are used.

It can be seen that in all cases the control responses have been improved drastically, especially the reduction in overshoot.

T_s	Initial Performance		Method	Final	Final Performance		
	$100M_p$	t_s		k_p, k_i, k_d	$100M_p$	t_s	N
0.01	39.16	64.14	M_0	3.56, 0.10, 21.62	0.13	14.77	896
			M_1	4.04, 0.12, 25.26	0.43	11.25	80
			M_2	3.24, 0.11, 21.12	1.12	17.64	43
0.05	39.52	63.90	M_0	3.45, 0.10, 20.89	0.057	15.45	768
			M_1	4.04, 0.12, 25.26	0.55	10.85	80
			M_2	3.25, 0.11, 21.12	1.10	17.35	43
0.2	40.92	63.00	M_0	3.53, 0.11, 21.76	0.060	13.60	1024
			M_1	4.08, 0.15, 28.21	1.02	11.80	96
			M_2	2.80, 0.090, 16.41	0.70	20.20	56
0.5	44.26	77.00	M_0	3.39, 0.11, 21.60	0.00	11.50	832
			M_1	3.48, 0.13, 24.05	0.97	15.50	72
			M_2	2.69, 0.086, 15.61	0.74	20.00	48
2	76.96	88.00	M_0	2.22, 0.082, 17.60	1.08	34.00	576
			M_1	2.19, 0.078, 15.22	0.32	34.00	128
			M_2	2.16, 0.075, 13.94	0.0020	18.00	81

Table 7.6: Digital Control Results. Initial performance is achieved by ZN tuned PID. Final performance is achieved by ILT.

7.5 Real-Time Implementation

In order to show the applicability and effectiveness of the proposed ILT, real-time experiment has been carried out on a couple tank.

7.5.1 Experimental setup and plant modelling

The couple tank equipment consists of two small perspex tower-type tanks and interconnected through an hole which yields a hydraulic resistance (Fig. 7.9). The experimental setup in this work was configured such that the level is measured from the tank-2 while the water is pumped into the tank-1 as the control input. The outlet of tank-2 is used to discharge the water into the reservoir. The measurement data is collected from couple tank using NI data acquisition card USB-6008. The control method is programmed using Labview. A window-type smoothing filter using 100 samples is implemented to mitigate measurement noise. The sampling period is 0.125 second.

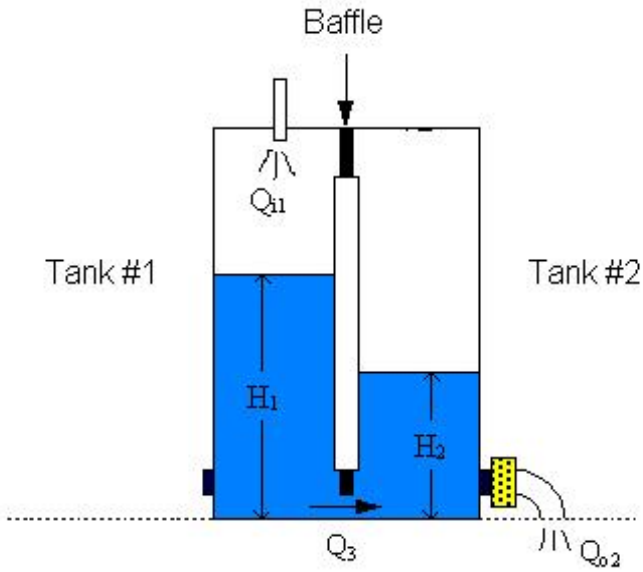


Figure 7.9: Diagram of couple tank apparatus

A step response is conducted to approximate the couple tank dynamics with a first order plus time delay model [10]

$$G(s) = \frac{k}{\tau s + 1} e^{-sL} \tag{7.15}$$

where k is the plant DC gain, τ is the time constant, L is the transportation lag. As shown in Fig. 7.10, after conducting a step response the obtained plant is

$$G(s) = \frac{168}{1 + 65s} e^{-3.6s}. \tag{7.16}$$

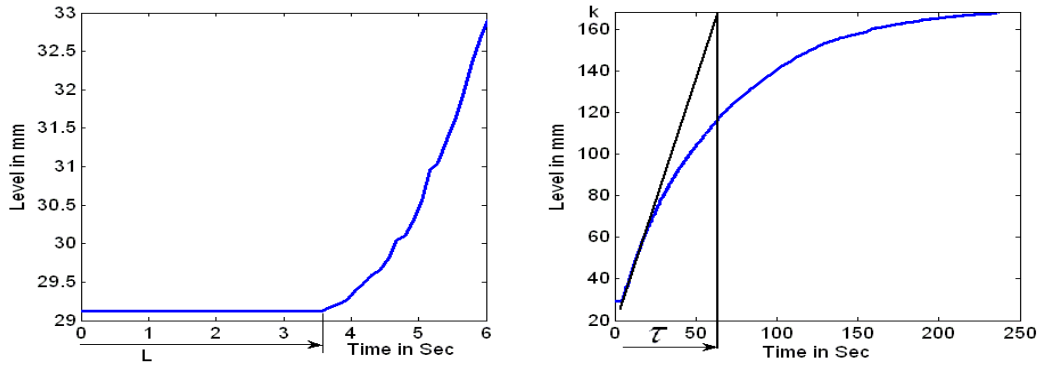


Figure 7.10: Step response based modelling

7.5.2 Application of ILT method

In the experiments, we conducts a series of tests to investigate ILT. The quadratic objective function is

$$J = (100M_p)^2 + qt_s^2$$

where two values $q = 0.1$ and $q = 0.01$ are used. The ISE is

$$\frac{1}{T - t_0} \int_{t_0}^{T} e^2 dt,$$

where $T = 124$ s and $t_0 = 27.4$ s. All three search methods M_0, M_1 and M_2 are applied.

It is observed in the experiment that the calculated gradient components (7.11) or (7.12) may be singular some times. This is due to the presence of measurement noise and the closeness of the values of J at two adjacent iterations. On the other hand, the learning process may become sluggish when the PID parameters at two adjacent iterations are too close, yielding a very lower learning gains $\gamma_{j,i}$. To solve these two problems, a constraint is applied when updating the learning gains

$$c_1|k_{j,i}| \leq \gamma_{j,i} \leq c_2|k_{j,i}|, \tag{7.17}$$

where $0 \leq c_1 < c_2 \leq 1$. The upper and lower learning gain bounds are made in proportion to PID parameters. The rationale is clear. When a controller parameter is bigger, we can update it with a larger bound without incurring drastic changes. If setting absolute bounds for the learning gain, an overly small bound would limit the parameter updating speed, and an overly large bound would make the constraint ineffective. In the real-time application we consider 2 sets of boundaries

C_1: $c_1 = 0.02 \quad c_2 = 0.2$,

C_2: $c_1 = 0.05 \quad c_2 = 0.4$.

The initial PID controller is found using ZN tuning method

$$0.11 + \frac{0.011}{s} + 0.26s.$$

The transient performance with ZN tuned PID is $100M_p = 30.02$ and $t_s = 58.73$ s. ILT is applied to improve the performance.

7.5.3 Experimental results

The results are summarized in Table 4.7. It can be seen that searching methods M_1 and M_2 can significantly reduce the number of learning iterations while M_p and t_s can be

Method	Constraint	J	PID Controller	$100M_p$	t_s	N
M_0	C_1	$(100M_p)^2 + 0.01t_s^2$	$0.11 + 0.0029s^{-1} + 0.26s$	0.089	16.65	384
M_1	C_1	ISE	$0.16 + 0.0045s^{-1} + 0.32s$	4.18	34.04	56
	C_2	ISE	$0.14 + 0.0039s^{-1} + 0.20s$	4.29	30.22	32
	C_1	$(100M_p)^2 + +0.01t_s^2$	$0.21 + 0.0036s^{-1} + 0.43s$	0.11	16.58	56
	C_2	$(100M_p)^2 + 0.01t_s^2$	$0.16 + 0.0041s^{-1} + 0.34s$	0.78	10.45	24
	C_1	$(100M_p)^2 + 0.1t_s^2$	$0.26 + 0.0037s^{-1} + 0.45s$	1.40	9.35	48
	C_2	$(100M_p)^2 + 0.1t_s^2$	$0.22 + 0.0036s^{-1} + 0.44s$	1.44	10.10	32
M_2	C_1	ISE	$0.26 + 0.0040s^{-1} + 0.62s$	1.30	10.50	15
	C_2	ISE	$0.23 + 0.0042s^{-1} + 0.56s$	2.19	12.98	13
	C_1	$(100M_p)^2 + 0.01t_s^2$	$0.32 + 0.0034s^{-1} + 0.76s$	0.27	17.30	17
	C_2	$(100M_p)^2 + 0.01t_s^2$	$0.30 + 0.0034s^{-1} + 0.72s$	0.50	16.63	14
	C_1	$(100M_p)^2 + 0.1t_s^2$	$0.30 + 0.0038s^{-1} + 0.72s$	1.30	16.33	14
	C_2	$(100M_p)^2 + 0.1t_s^2$	$0.23 + 0.0037s^{-1} + 0.56s$	1.71	15.99	10

Table 7.7: Experimental Results.

maintained at almost the same level. By changing the weight for t_s, the final overshoot M_p and settling time t_s can be adjusted. The constraint C_2 can also reduce the trial number, because higher limits in learning gains will expedite the learning progress. In experiments, it is observed that M_2 can be further simplified by removing the last run of exhaustive searching, so long as the variation of J reaches the preset threshold ε.

7.6 Conclusion

A new PID auto-tuning method is developed and compared with several well established auto-tuning methods including ZN, IFT, ES, PPT, IMC. Iterative learning tuning provides a sustained improvement in closed-loop control performance, and offers extra degrees of freedom in specifying the transient control requirements through a new objective function.

The iterative learning tuning method proposed in this work can be further extended to various optimal designs for controllers, owing to its model free nature. For instance, by iterative searching and task repeatability, we can tune the parameters of lead and lag compensators, filters, observers, and use both time domain and frequency performance indices in objective functions.

Chapter 8

Conclusions

8.1 Summary of Results

In Chapters 2, and 3, the parametric adaptive control of nonlinear systems are considered, where the uncertainties are state-periodic in the first system and time-periodic in the other system.

Specifically, in Chapter 2, a spatial periodic adaptive control approach is proposed to deal with nonlinear rotary machine systems with a class of state-varying parametric uncertainties. Since the parametric uncertainties are not time-periodic, more difficulties are encountered in the process of controller design. By focusing on the relationship between the systems with state-dependent uncertainties and the systems with time-dependent uncertainties, the system is converted to a more complex one while possessing time-periodic parameters as uncertainties. As a result, by applying those known achievements for time-peridoic systems, our difficulty is overcome properly in spatial control for the plant with highly nonlinear components.

As another theoretical development, in Chapter 3, we develop a general discrete-time adaptive control approach suitable for nonlinear systems with periodic parametric uncertainties. The underlying idea of the new approach is to convert the periodic parameters

into an augmented constant parametric vector by a lifting technique. The novelty of this approach is the establishment of a bridge between classical adaptive control problems and periodic adaptive control problems. As such, the well-established discrete-time adaptive control schemes can be easily applied to various control problems with periodic parameters, such as plants that do not meet the linear growth condition, plants that are nonlinear in parameters, plants with unknown control directions, plants in parametric-strict-feedback form, etc. Another major advantage of the new adaptive control is the ability to adaptively update all parameters in parallel, hence expedite the adaption speed.

In Chapters 4, 5, 6, and 7, we apply the central idea of ILC to the learning of initial system state, the learning of input nonlinearity, the learning of boundary condition for PDE processes, and the tuning of PID parameters respectively. In all the four topics, the system control process is always assumed to be strictly repeatable.

Firstly, an initial state ILC approach is proposed for final state control of motion systems. ILC is applied to learn the desired initial states in the presence of system uncertainties. Four cases are considered where the initial position or speed are manipulated variables and final displacement or speed are controlled variables. In these cases, the motion system could have discontinuous damping or discontinuous frictions but Lipschitzian in position. By duality, we further explore other four cases if the motion system is Lipschitz continuous in speed. Since the control task is specified spatially in states, a state transformation is introduced such that the final state control problems are formulated in the phase plane to facilitate spatial ILC design and analysis.

In Chapter 5, a dual-loop ILC scheme is designed for a class of nonlinear systems with hysteresis input uncertainty. The two ILC loops are applied to the nominal part and the hysteresis part respectively, to learn their unknown dynamics. Based on the

convergence analysis for each single loop, a composite energy function method is then adopted to prove the learning convergence of the dual-loop system in iteration domain. When the strict input-output monotonicity does not exist in the hysteretic loop, the ILC law is revised by adding a forgetting factor and incorporating a time-varying learning gain, and then ensure the corresponding ILC operator to be contractible. By using the Banach fixed-point theorem, we show that the output tracking error of the inner ILC loop and then the dual ILC loop can enter and remain ultimately in a small neighborhood of zero.

In Chapter 6, we further apply the idea of ILC to the velocity boundary control of a class of quasi-linear PDE processes. When the whole process is strictly repeatable in iteration domain and the steady state output on the other boundary is concerned only, we simplified the system plant from PDE to ODE based on several important assumptions. The advantages of the proposed controller are its simple structure, strict convergence ensurance, and capability of dealing with input saturations and measurement delays easily.

At last, an optimal tuning method for PID parameters is proposed by means of iterative learning. In the scheme, the time domain performance or requirements are incorporated directly into the objective function, and then the problem of PID tuning is converted to minimizing the function in an iterative manner. By formulating it as an iterative learning process, the optimal tuning does not require as much the plant model knowledge as other PID tuning methods, and can be applied straightforward to discrete-time or sampled-data systems, in contrast to existing PID auto-tuning methods which are dedicated to continuous-time plants. In this chapter, through theoretical analysis, comprehensive investigations on benchmarking examples, and real-time experiments on

the level control of a coupled-tank system, the effectiveness of the proposed method is validated.

8.2 Suggestions for Future Work

Past research activities have laid a foundation for the future work. Based on the prior research, the following questions deserve further consideration and investigation.

1. The models discussed in the thesis are assumed to be precise. Unfortunately repeatable or non-repeatable noise cannot be omitted in the application. In order to improve our controller design, it is inevitable to take the system noise into consideration.

2. Up to present, the spatial periodic adaptive control is considered only for systems with canonical form. But in practice, the system dynamics could take a more general structure, e.g., the cascaded form. In the future work, extensions should be done for those general ones which could possess state-varying uncertainties.

3. More scenarios should be considered in the dual-loop ILC design, including (1) the model is of MIMO form, (2) the state of the hysteresis output is not measureable, (3) other types of hysteresis model, and (4) the input uncertainties take a more complex structure, e.g., a combination of deadzone, saturation, and hysteresis.

4. The planar requirement for the motion system is important in the initial state ILC. Without this requirement, the monotonicity and uniqueness of solution can not help us fully solve the final state control task. So, one remaining problem is how to extend the result in Chapter 4 to higher-order motion systems.

5. In all the proposed parametric adaptive control methodologies, the learning ability for parameters does not guarantee that their values can be achieved after the learning process. Whether a revised adaptive learning scheme can be found to compensate for

the pity is a highly challenging problem.

6. In Chapter 6, only the set-point problem was conerned in the boundary control of PDE processes. It is desirable to consider more general control problems for such a plant, e.g. periodic tracking problem, and explore control schemes for other more complex PDE processes.

7. Besides these points, more control problems with repetitiveness, which can be formulated or solved with learning-type control strategies, stimulate our research in the near future. They include stochastic learning in multi-agent systems, adaptive learning in vibration control of hard disk drive, etc.

Bibliography

[1] K. Abidi and J.-X. Xu, "A discrete-time periodic adaptive control approach for time-varying parameters with known periodicity", *IEEE Trans. Automat. Control*, Vol. 53, No. 2, pp.575-581, 2008.

[2] H. S. Ahn, K. L. Moore, and Y. Q. Chen, *Iterative learning control: robustness and monotonic convergence in the iteration domain*, Springer-Verlag, Communications and Control Engineering Series, ISBN: 978-1-84628-846-3, 2007.

[3] H. S. Ahn and Y. Q. Chen, "Periodically time varying adaptive friction compensation", In *Proc. 2004 IEEE International Conference on Robotics and Biomimetics (RoBio'04)* , Marriott, Shenyang, China, Aug. 22-25, 2004.

[4] H. S. Ahn, Y. Q. Chen and W. Yu, "Periodic adaptive compensation of state-dependent disturbance in a digital servo motor system", *International Journal of Control, Automation, and Systems*, Vol. 5, No. 3, pp.343-348, 2007.

[5] H. S. Ahn, Y. Q. Chen and H. F. Dong, "State-periodic adaptive compensation of cogging and coulomb friction in permanent-magnet linear motors", *IEEE Trans. on Magnetics*, Vol. 41, No. 1, pp.90-98, 2005.

[6] H. S. Ahn, Y. Q. Chen and Z. M. Wang, "State-dependent disturbance compensation in low-cost wheeled mobile robots using periodic adaptation", In *Intelligent Robots*

and Systems, 2005 (IROS 2005), 2005 IEEE/RSJ International Conference on 2-6 Aug., PP.729-934, 2005.

[7] H. S. Ahn, Y. Q. Chen, and K. L. Moore, "Iterative learning control: brief survey and categorization", *IEEE Trans. Systems, Man and Cybernetics-Part C: Applications and Reviews* , Vol. 37, Nn. 6, pp.1099-1121, 2007.

[8] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning", *J. Robot. Syst.*, Vol. 1, No. 2, pp.123-140, 1984.

[9] S. Arimoto, S. kawamura and F. Miyazaki, "Iterative learning control for robot systems", In *Proc. IECON*, Tokyo, Japan, pp.127-134, 1984.

[10] K. J. Astrom and T. Hagglund, *PID Controllers: Theory, Design and Tuning*, 2nd ed. Research Triangle Park, NC: Instrum. Soc. Amer., 1995.

[11] K. J. Astrom, T. Hagglund, C. C. Hang, and W. K. Ho, "Automatic tuning and adaptation for PID controllers – a survery", *Control Engineering Practice*, Vol. 1, no. 4, pp.699-714, 1993.

[12] K. J. Astrom and T. Hagglund, "Automatic Tuning of simple regulators with specifications on phase and amplitude margins", *Automatica*, Vol. 20, no. 5, pp.645-651, 1984.

[13] M. M. Belsnes, J. Roynstrand, and O. B. Fosso, "Handling state dependent nonlinear tunnel flows in short-term hydropower scheduling", In *Int. Conf. Power System Technology*, Singapore, 2004.

[14] J. Bentsman and Y. Orlov, "Reduced spatial order model reference adaptive control of spatially varying distributed parameter systems of parabolic and hyperbolic types", *Int. J. Adaptive Control and Signal Processing*, Vol. 15, pp.679-696, 2001.

[15] S. P. Bhat, and D. K. Min, "Precise point-to-point positioning control of flexible structure", *Trans. ASME J. Dyn. Syst. Meas. Control*, Vol. 112, No. 4, pp.667-674, 1990.

[16] Z. Bien and J.-X. Xu, *Iterative Learning Control - Analysis, Design, Integration and Applications*. Boston: Kluwer Academic Press, USA, 1998.

[17] D. M. Boskovic and M. Krstic, "Stabilization of a solid propellant rocket instability by state feedback", *Int. J. Robust Nonlin. Control*, Vol. 13, pp.483-495, 2003.

[18] A.-H. Brian, D. Pierreand, and C. D. W. Carlos, "A survey of models, analysis tools and compensation methods for the control of machines with friction", *Automatica*, Vol. 30, No. 7, pp.1083-1138, 1994.

[19] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A Survey of Iterative Learning Control", *IEEE Control systems Magazine*, Vol. 26, No. 3, pp.96-114, 2006.

[20] M. Brokate and A. Visintin, "Properties of the Preisach model for hysteresis", *J. Angew. Math.*, Vol. 402, pp.1-40, 1989.

[21] M. Brokate and J. Sprekels, *Hysteresis and phase transitions*, New York: Springer, 1996.

[22] I. V. Burkov and A. T. zaremba, "Adaptive control for angle speed oscillations generated by periodic disturbances", In *Proc. 6th St. Petersburg Symp. on Adaptive Systems Theory*, pp.34-36, St. Petersburg, 1999.

[23] R. H. S. Carpenter, *Neurophysiology*, Arnold London, 2003.

[24] C. D. W. Carlos and P. Laurent, "Adaptive eccentricity compensation", In *Proc. 37th IEEE Conference on Decision and Control*, pp.2271-2276, Tampa, Florida, Dec., 1998.

[25] P. Castilla, M. Meraz, O. Monroy, and A. Noyola, "Anaerobic treatment of low concentration waste water in an inverse fluidized bed reactor", Water Sci. Technol., Vol. 41, pp.245-251, 2000.

[26] C. D. W. Carlos and P. Laurent, "Adaptive eccentricity compensation", *IEEE Trans. on Control Systems Technology*, Vol. 8, No. 5, pp.757-766, 2000.

[27] J. M. Carrasco, E. Galvan, G. E. Valderrama, R. Ortega, and A. M. Stankovic, "Analysis and experimentation of nonlinear adaptive controllers for the series resonant converter", *IEEE Trans. Power Electron.*, Vol. 15, No. 3, pp.536-544, 2000.

[28] R. A. Calico and W. E. Wiesel, "Stabilizaion of helicopter blade flapping", *J. American Helicopter Society*, Vol. 31, pp.59-64, 1986.

[29] Y. Chen, C. Wen, Z. Gong and M. Sun, "An iterative learning controller with initial state learning", *IEEE Trans. Automat. Control*, Vol. 44, No. 2, pp.371-376, 1999.

[30] Z. Chen and J. Huang, "A simplified small gain theorem for time-varying nonlinear systems", *IEEE Trans. Automat. Control*, Vol. 50, No. 11, pp.1904-1908, 2005.

[31] J. Cole and R. Calico, "Nonlinear oscillations of a controlled periodic system", *Control Engineering Practice*, Vol. 15, pp.627-633, 1992.

[32] B. J. Driessen and N. Sadegh, "Multi-input square iterative learning control with input rate limits and bounds", *IEEE Trans. Syst. Man Cybern. B: Cybern.*, Vol. 32, No. 4, pp.545-550, 2002.

[33] A. O. Dwyer, *Handbook of PI and PID Controller Tuning Rules*, Imperial College Press, London, 2003.

[34] D. Dochain, J. P. Babary, and N. Tali-Maamar, "Modelling and adaptive control of nonlinear distributed parameter bioreactors via orthogonal collocation", *Automatica*, Vol. 28, pp.873-883, 1992.

[35] G. Feng and R. Lozano, *Adaptive Control Systems*, Oxford, U.K.: Newnes, 1999.

[36] B. Fidan, Y. Zhang, and P. A. Ioannou, "Adaptive control of a class of slowly time varying systems with modeling uncertainties", *IEEE Trans. Automat. Control*, Vol. 50, pp.915-920, 2005.

[37] G. F. Froment and K. B.Bischoff, *Chemical Reactor Analysis and Design*, Wiley, New York, 1990.

[38] G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction, and Control*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[39] S. S. Ge, C. Yang, and T. H. Lee, "Adaptive robust control of a class of nonlinear strict-feedback discrete-time systems with unknown control directions", *Systems & Control Letters*, Vol. 57, No. 11, pp.888-895, 2008.

[40] S.-H. Han, Y.-H. Kim, and I.-J. Ha, "Iterative identification of state-dependent disturbance torque for high-precision velocity control of servo motors", *IEEE Trans. Automat. Control*, Vol. 43, No. 5, pp.724-729, 1998.

[41] S. Hara, Y. Yamamoto, T. Omata and M. Nakano, "Repetitive control system: a new type servo system for periodic exogenous signals", *IEEE Trans. Automat. Control*, Vol. 33, No. 7, pp.659-668, 1988.

[42] S. Haykin, *Neural Networks, A comprehensive Foundation*, Prentice Hall, Upper Saddle River, New Jersey, 1994.

[43] R. Hull, C. Ham. and R. Johnson, "Systematic design of attitude control systems for a satellite in a circular orbit with guaranteed performance and stability", In *Proc. AIAA/USU Conference on Small Satellite*, Logan, UT, USA, Aug. 2000.

[44] D. Huang, Y. Tan, and J.-X. Xu, "A Dual-loop Iterative Learning Control for Nonlinear Systems with Hysteresis Input Uncertainty", Submitted.

[45] D. Huang, J.-X. Xu, and Z. Hou, "A Discrete-Time Periodic Adaptive Control Approach for Parametric-strict-feedback Systems", In *IEEE Conf. on Decision and Control*, December, Shanghai, 2009.

[46] D. Huang, Y. Tan, and J.-X. Xu, "A Dual-loop Iterative Learning Control for Nonlinear Systems with Hysteresis Input Uncertainty", In *7th IEEE Int. Conf. on Control & Automation (ICCA09)*, December, New Zealand, 2009.

[47] G. Hillerstrom and K. Walgama, "Repetitive control theory and applications- a survey", In *Proc. 13th World Congress Vol. D: Control Design II, Optimization*, pp.1-6, 1997.

[48] W. K. Ho, C. C. Hang, and L. S. Cao, "Tuning of PID controllers based on gain and phase margin specifications", *Automatica*, Vol. 31, No. 3, pp.497-502, 1995.

[49] W. K. Ho, O. P. Gan, E. B. Tay, and E. L. Ang, "Performance and gain and phase margins of well-known PID tuning formulas", *IEEE Trans. Control Systems Technology*, Vol. 4, No. 3, pp.473-477, 1996.

[50] R. Holsapple, R. Venkataraman, and D. Doman, "A modified simple shooting method for solving two-point boundary-value problems", In *Proc. 2003 IEEE Aerospace Conference*, pp.2783-2790, 2003.

[51] K.-S. Hsu, "Stability and analysis of iterative learning system using frequency domain method", *Int. J. Computer Math.*, Vol. 82, No. 11, pp.1339-1353, 2005.

[52] M. Hu, H. Du, S.-F. Ling, Z. Zhou, and Y. Li, "Motion control of an electrostrictive actuator", *Mechatronics*, Vol. 14, No. 2, pp.153-161, 2004.

[53] T. Hu, D. Wang, L. Shen, Y. Sun, and H. Wang, "Iterative learning control for a class of systems with hysteresis", In *Proc. 10th Int. Conf. Control, Automation, Robotics, and Vision*, Hanoi, Vietnam, Dec 17-20, pp.10-15, 2008.

[54] G. Hua and M. Sun, "Learning identification and control of a class of discrete periodic systems", In *Proc. 4th IEEE Conf. on Industrial Electronics and Applications*, Xi'an, China, pp.6-11, May 25-27, 2009.

[55] M. Ikeda and D. D. Siljak, "Optimality and robustness of linear-quadratic control for nonlinear systems", *Automatica*, Vol. 26, no. 3, pp.499-511, 1990.

[56] F. Ikhouane, V. Mañosa, and J. Rodellar, "Dynamic properties of the hysteretic Bouc-Wen model", *Systems and Control Letters*, Vol. 56, pp.197-205, 2007.

[57] F. Ikhouane and J. Rodellar, "On the hysteretic Bouc-Wen model Part I: Forced limit cycle characterization", *Nonlinear Dynamics*, Vol. 42, pp.63-78, 2005.

[58] F. Ikhouane and J. Rodellar, "On the hysteretic Bouc-Wen model Part II: Robust parametric identification", *Nonlinear Dynamics*, Vol. 42, pp.79-95, 2005.

[59] F. Ikhouane, J. E. Hurtado, and J. Rodellar, "Variation of the hysteresis loop with the Bouc-Wen model parameters", *Nonlinear Dyn.*, Vol. 48, pp.361-380, 2007.

[60] P. A. Ioannou and J. Sun, *Robust Adaptive Control*, Upper Saddle River, NJ: Prentice-Hall, 1996.

[61] R. Isniewski and M. Blanke, "Fully magnetic attitude control for spacecraft subject to gravity gradient", *Automatica*, Vol. 35, pp.1201- 1214, 1999.

[62] M. A. Johnson and M. H. Moradi, *PID Controller Design*, Spring-Verlag London, 2003.

[63] A.-R. Jose, P. Hector, and J. O.-T. Alberto, "Linear boundary control for a class of nonlinear PDE processes", *Systems & Control Letters*, Vol. 44, pp.395-403, 2001.

[64] K. Kaneko and R. Horowitz, "Repetitive and adaptive control of robot manipulators with velocity estimation" , *IEEE Trans. Robot. Automat.*, Vol. 13, No. 2, pp.204-217, 1997.

[65] I. Kanellakopoulos, "A discrete-time adaptive nonlinear system", *IEEE Trans. Automat. Control*, Vol. 39, No. 11, pp.2362-2365, 1994.

[66] N. J. Killingsworth and M. Krstic, "PID tuning using extremum seeking: online, model-free performance optimization", *IEEE Control Systems Magazine*, Vol. 26, no. 1, pp.70-79, 2006.

[67] K. P. V. Kokotovic, *Fundations of Adaptive Control*, New York: Springer-Verlag, 1991.

[68] M. A. Krasnolesl'skii and A. V. Pokrovskii, *Systems with hysteresis*, Springer-Verlag, 1989.

[69] M. Krstic and A. Smyshlyaev, "Backstepping boundary control for first-order hyperbolic PDEs and application to systems with actuator and sensor delays", *Syst. Control Lett.*, Vol. 57, pp.750-758, 2008.

[70] C. Y. Kuo, C. L. Yang, and C. Margolin, "Optimal controller design for nonlinear systems", *IEE Proc. on Control Theory and Applications*, Vol. 145, no. 1, pp.97-105, 1998.

[71] K. K. Leang and S. Devasia, "Design of hysteresis-compensating iterative learning control for piezo-positioners: Application to atomic force microscopes", *Mechatronics*, Vol. 16, pp.141-158, 2006.

[72] K. K. Leang, S. Ashley, and G. Tchoupo, "Iterative and feedback control for hysteresis compensation in SMA", *J. Dynamic Systems, Measurement, and Control*, Vol. 131, No. 1, p014502 (6 pp.), 2009.

[73] K. K. Leang and S. Devasia, "Iterative feedforward compensation of hysteresis in piezo positioners", In *Proc. IEEE 42nd conf. decision and control*, Maui, Hawaii, USA, Dec., pp.2626-2631, 2003.

[74] T. H. Lee and K. S. Narendra, "Stable discrete adaptive control with unknown high-frequency gain", *IEEE Trans. Automat. Control*, Vol. 31, No. 5, pp.477-478, May 1986.

[75] O. Lequin, E. Bosmans, and T. Triest, "Iterative feedback tuning of PID parameters: comparison with classical tuning rules", *Control Engineering Practice*, Vol. 11, no. 9, pp.1023-1033, 2003.

[76] A. Leva, "PID autotuning algorithm based on relay feedback", *IEEE Proc. D Contr. Theory Applicat.*, Vol. 140, no. 5, pp.328-338, 1993.

[77] K. K. Leang and S. Devasia, "Iterative feedforward compensation of hysteresis in piezo positioners" *In IEEE 42nd conference on decision and controls*, Maui, Hawaii, USA, pp.2626-2631, December, 2003.

[78] C. S. Liu and H. Peng, "Disturbance observer based tracking control", *Trans. of ASME*, Vol. 122, pp.332-335, 2000.

[79] L. Liu, K. K. Tan, A. S. Putra, and T. H. Lee, "Compensation of hysteresis in piezoelectric actuator with iterative learning control", In *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, Suntec Convention and Exhibition Center, Singapore, July 14-17, pp.1300-1305, 2009.

[80] A. P. Loh, C. Y. Qu, and K. F. Fong, "Adaptive control of discrete time systems with concave/convex parametrizations", *IEEE Trans. Automat. Control*, Vol. 48, No. 6, pp.1069-1072, 2003.

[81] R. W. Longman, "Iterative learning control and repetitive control for engineering practice", *Int. J. Control*, Vol. 73, No. 10, pp.930-954, 2000.

[82] G. K. I. Mann, B.-G. Hu, and R. G. Gosine, "Time-domain based design and analysis of new PID tuning rules", In *IEE Proc. Control Theory and Applications*, Vol. 148, no. 3, pp.251-261, 2001.

[83] F. S. Mats, T. T. Hannu, and L. Bengt, "State-space solution to the periodic multirate H_∞ control problem: a lifting approach", *IEEE Trans. Automat. Control*, Vol. 45, No. 12, pp.2345-2349, Dec. 2000.

[84] F. Mazen, L. B. Carolyn, and E. D. Geir, "Model reduction of periodic systems: a lifting approach", *Automatica*, Vol. 41, pp.1085-1090, 2005.

[85] W. Messner, R. Horowitz, W. W. Kao, and M. Boa, "A new adaptive learning rule", *IEEE Trans. Automat. Control*, Vol. 36, No. 2, pp.188-197, 1991.

[86] C. Mi, H. Lin, and Y. Zhang, "Iterative learning control of antilock braking of electric and hybrid vehicles", *IEEE Trans. on Vehicular Technology*, Vol. 54, No. 2, pp.486-494, 2005.

[87] S. Mishra, U. Topcu, and M. Tomizuka, "Iterative learning control with saturation constraints", In *Proc. 2009 American Control Conf.*, Hyatt Regency Riverfront, St. Louis, MO, USA, June 10-12, pp.943-948, 2009.

[88] K. L. Moore, *Iterative learning control for deterministic systems*, Vol. 22 of *Advances in Industrial Control*, Springer-Verlag, London.

[89] K. L. Moore, "Iterative learning control - an expository overview", *Applied & Computational Controls, Signal Processing, and Circuits*, Vol. 1, No. 1, pp.1-42, 1998.

[90] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, Upper Saddle River, NJ: Prentice-Hall, 3, 1989.

[91] K. S. Narendra and Z. Tian, "Adaptive control of linear periodic systems", In *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, California, pp.465-470, 2006.

[92] Y. Q. Ni, J. M. Ko, and C. W. Wong, "Identification of non-linear hysteretic isolators from periodic vibration tests", *J. sound and Vibration*, Vol. 217, No. 4, pp.737-756, 1998.

[93] H. Nishimura, and K. Funaki, "Motion control of three-link brachiation robot by using final-state control with error learning", *IEEE/ASME Transactions on Mechatronics*, Vol. 3, No. 2, pp.120-128, 1998.

[94] D. H. Owens, E. Rogers and K. Galkowski, "Control theory and applications for repetitive processes", *Advances in Control, Highlights of ECC'99*, Vol. 2, No. 3, pp.327-333, 1999.

[95] P. R. Pagilla, B. Yu, and K. L. Pau, "Adaptive control of time-varying mechanical systems: Analysis and experiments", *IEEE/ASME Trans. Mechatron.*, Vol. 5, No. 4, pp.410-418, 2000.

[96] Z. H. Qu, "Robust control of nonlinear systems by estimating time variant uncertainties", *IEEE Trans. Automat. Contr.*, Vol. 47, No. 1, pp.115-121, 2002.

[97] Z. H. Qu and J.-X. Xu, "Model based learning control and their comparisons using Lyapunov direct method", *Asian J. Control*, Vol. 4, pp.99-110, 2002.

[98] W. H. Ray, "Some recent applications of distributed parameter systems theory-a survey", *Automatica*, Vol. 14, pp.281-287, 1978.

[99] B. Ren, S. S. Ge, T. H. Lee, and C.-Y. Su, "Adaptive neural control of SISO non-affine nonlinear time-delay systems with unknown hysteresis input", In *Proc. 2008 American Control Conf.*, Seattle, WA, pp.4203-4208, June 11-13, 2008.

[100] M. Saeki, "Unfalsified control approach to parameter space design of PID controllers", In *Proc. 42nd IEEE Conf. on Decision and Control*, pp.786-791, 2003.

[101] S. S. Sastry and M. Boston, *Adaptive Control: Stability, Convergence and Robustness*, Upper Saddle River, NJ: Prentice-Hall, 1989.

[102] R. Salomon, "Evolutionary algorithms and gradient search: similarities and differences", *IEEE Trans. On Evolutionary Computation*, Vol. 2, no. 2, pp.45-55, 1998.

[103] R. Salomon and J. L. van Hemmen, "Accelerating backpropagation through dynamic self-adaptation", *Neural Networks*, Vol. 9, no. 4, pp.589-601, 1996.

[104] J. Seo, R. Venugopal and J.-P. Kenne, "Feedback linearization based control of a rotational hydraulic drive," *Int. J. Control, Automation, and Systems*, Vol. 5, No. 3, pp.343-348, 2007.

[105] A. W. Smyth, S. F. Masri, A. G. Chassiakos, and T. K. Caughey, "On-line parametric identification of MDOF nonlinear hysteretic systems", *J. Engineering Mechanics*, Vol. 125, No. 2, pp.133-142, 1999.

[106] Z. Shareefdeen and B. C. Baltzis, "Biofiltration of toluence vapor under steady-state and transient conditions: theory and experimental results", *Chem. Eng. Sci.*, Vol. 49, pp.4347-4360, 1994.

[107] A. Smyshlyaev and M. Krstic, "Lyapunov adaptive boundary control for parabolic PDEs with spatially varying coefficients", in *Proc. Amer. Control Conf.*, pp.41-48, 2006.

[108] A. Smyshlyaev and M. Krstic, "Adaptive boundary control for unstable parabolic PDEs-Part II: Estimation-based designs", *Automatica*, Vol. 43, pp.1543-1556, 2007.

[109] A. Smyshlyaev and M. Krstic, "Adaptive boundary control for unstable parabolic PDEs-Part III: Output-feedback examples with swapping identifiers", *Automatica*, Vol. 43, pp.1557-1564, 2007.

[110] V. Solo and B. Bamieh, "Adaptive distributed control of a parabolic system with spatially varying parameters", *in Proc. 38th IEEE Conf. Dec. Control*, pp.2892-2895, 1999.

[111] A. Smyshlyaev and M. Krstic, "Backstepping observers for a class of parabolic PDEs", *Syst. Control Lett.*, Vol. 54, pp.613-625, 2005.

[112] J. Song and A. D. Kiureghian, "Generalized Bouc-Wen model for highly asymmetric hysteresis", *J. Engineering Mechanics-ASCE*, Vol. 132, No. 6, pp.610-618, 2006.

[113] B. Straughan, *The Energy Method, Stability, and Nonlinear Convection*, Springer, New York, 1992.

[114] T. Sugie and T. Ono, "An iterative learning control law for dynamical systems", *Automatica*, Vol. 27, No. 4, pp.729-732, 1991.

[115] M. X. Sun and D. W. Wang, "Itertive learning control with initial rectifying action", *Automatica*, Vol. 38, pp.1177-1182, 2002.

[116] K. K. Tan, Q. G. Wang, C. C. Hang and T. Hagglund, *Advances in PID Control*, Advances in Industrial Control Series, Springer Verlag: London, 1999.

[117] Y. Tan and J.-X. Xu, "A note on iterative learning control for nonlinear systems with input uncertainties", In *Proc. 17th Int. Federation of Automatic Control*, July 06-11, Seoul, Korea, 2008.

[118] G. Tao and P. V. Kokotović, "Adaptive control of plants with unknown hysteresis", *IEEE Trans. Automatic Control*, Vol. 40, No. 2, pp.200-212, 1995.

[119] G. Tchobanoglous and F. Burton, *Wastewater Engineering- Treatment, Disposal and Reuse*, 3rd Edition, McGraw-Hill Series in Water Resources and Environmental Engineering, Metcalf & Eddy, Inc., New York, 1991.

[120] Z. Tian and K. S. Narendra, "Adaptive control of linear periodic systems", In *Proc. American Control Conference*, Hyatt Regency Riverfront, St. Louis, MO, USA, pp.1274-1279, June 10-12, 2009.

[121] Uwe Kiencke, and Lars Nielsen, *Automotive Control Systems*, Berlin : New York : Springer-Verlag ; Warrendale, PA : SAE International, 2000.

[122] V. R. Vemuri, *Artificial Neural Networks: Concepts and Control Applications*, IEEE Comput. Society Press, 1995.

[123] Visintin, A., *Mathematical models of hysteresis*, Topics in nonsmooth analysis, Birkhaser Verlag, pp.295-326, 1988.

[124] A. A. Voda and I. D. Landau, "A method for the auto-calibration of PID controllers", *Automatica*, Vol. 31, no.1, pp.41-53, 1995.

[125] R. Venkataraman and P. Krishnaprasad, "Approximate inversion of hysteresis: theory and numerical results", In *Proc. 39th IEEE conf. on decision and control*, Sydney, Australia, pp.4448-4454, December, 2000.

[126] Q. Wang, T. Lee, H. Fung, Q. Bi and Y. Zhang, "ID tuning for improved performance", *IEEE Trans. on Control Systems Technology*, Vol. 7, No. 4, pp.457-465, 1999.

[127] Z. Wang, M. M. Polycarpou, J. G. Uber, and F. Shang, "Adaptive control of water quality in water distribution networks", *IEEE Trans. Control Syst. Technol.*, Vol. 14, No. 1, pp.149-156, 2006.

[128] C. Wei and L. Guo, "Prediction-based discrete-time adaptive non-linear stochastic control", *IEEE Trans. Automat. Control*, Vol. 44, pp.1725-1729, 1999.

[129] C. Wei and Z. Chen, "Prediction-based discrete-time stochastic adaptive control: a new step towards non-linear growth rate", *Int. J. Control*, Vol. 79, pp.1002-1009, 2006.

[130] C. Xiang, "Adaptive control of discrete-time systems using multiple models", *PdD thesis*, Yale University, May 2000.

[131] Y. Xie, Y. Wang, L. Chen and C. H. Mastrangelo, "Fourier microfluidics", *Lab on a chip*, Vol. 8, pp.779-785, 2008.

[132] J.-X. Xu and Y. Tan, "On the p-type and newton-type ILC schemes for dynamic systems with non-affine-in-input factors", *Automatica*, Vol. 38, pp.1237-1242, 2002.

[133] J.-X. Xu and Y. Tan, "A composite energy function based sub-optimal learning control approach for nonlinear systems with time-varing parametric uncertainties", In *Proc. 39th IEEE Conf. on Design and Control*, Sydney, Australia, pp.3837-3842, 2000.

[134] J.-X. Xu, "The frontiers of iterative learning control-part II", *J. Systems, Control and Information*, Vol. 46, No. 5, pp.233-243, 2002.

[135] J.-X. Xu and J. Xu, "Iterative learning control for non-uniform trajectory tracking problems", In *Proc. 15th World Congress of IFAC*, Barcelona, Spain, 2002.

[136] J.-X. Xu, V. Badrinath, and Z. H. Qu, "Robust learning control for robotic manipulators with an extension to a class of non-linear systems", *Int. J. Control*, Vol. 73, pp.858-870, 2000.

[137] J.-X. Xu and Y. Tan, "A suboptimal learning control scheme for nonlinear systems with time-varing parametric uncertainties", *J. Optimal Control-Applications and Theory*, 2001.

[138] J.-X. Xu, "A New periodic adaptive control approach for time-varying parameters with known periodicty", *IEEE Trans. Automat. Control*, Vol. 49, No. 4, pp.579-583, 2004.

[139] J.-X. Xu, "Direct learning of control efforts for trajectories with different time scales", *IEEE Trans. Automat. Control*, Vol. 43, No. 7, pp.1027-1030, 1998.

[140] J.-X. Xu, S. K. Panda, Y. J. Pan, T. H. Lee and B. H. Lam, "Amodular control scheme for PMSM speed control with pulsating torque minimization", *IEEE Trans. on Industrial Electronics*, Vol. 51, No. 3, pp.526-536, 2004.

[141] J.-X. Xu and R. Yan, "Repetitive learning control: A time-delay approach for systems with periodic componets", Submitted.

[142] J.-X. Xu and R. Yan, "Synchronization of chaotic systems via learning control", *Int. J. Bifurcation and Chaos*, Accepted.

[143] J.-X. Xu, Y. Tan, and T. H. Lee, "Iterative learning control design based on composite energy function with input saturation", *Automatica*, Vol. 40, pp.1371-1377, 2003.

[144] J.-X. Xu, Y. Tan, and T. H. Lee, "Iterative learning control design based on composite energy function with input saturation", In *Proc. 2003 American Control Conf.*, Denver, Colorado, June 4-6, pp.5129-5134, 2003.

[145] J.-X. Xu, J. Xu, and T. H. Lee, "Iterative learning control for systems with input deadzone", *IEEE Trans. Automat. Control*, Vol. 50, No. 9, pp.1455-1459, 2005.

[146] J.-X. Xu, J. Xu, and T. H. Lee, "Iterative learning control for systems with input deadzone", In *Proc. 43rd IEEE Conf. Decision and Control*, Vol. 2, Dec 14-17, pp.1307-1312, 2004.

[147] J.-X. Xu, Y. Chen, T. H. Lee, and S. Yamamoto, "Terminal iterative learning control with an application to RTPCVD thickness control", *Automatica*, Vol. 35, No. 9, pp.1535-1542, 1999.

[148] J.-X. Xu, and Y. Tan, *Linear and Nonlinear Iterative Learning Control*, Berlin: Springer-Verlag, 2003.

[149] J.-X. Xu and R. Yan, "On repetitive learning control for periodic tracking tasks", *IEEE Trans. Automat. Control*, Vol. 51, No. 11, pp.1842-1848, 2006.

[150] J.-X. Xu, S. K. Panda, Y. J. Pan, T. H. Lee, and B. H. Lam, "A modular control scheme for PMSM speed control with pulsating torque minimization", *IEEE Trans. Ind. Electron.*, Vol. 51, No. 3, pp.526-536, 2004.

[151] J.-X. Xu and R. Yan, "Fixed point theorem-based iterative learning control for LTV systems with input singularity", *IEEE Trans. Automat. Control*, Vol. 48, No. 3, pp.487-492, 2003.

[152] J.-X. Xu and D. Huang, "Spatial Periodic Adaptive Control for Rotary Machine Systems", *IEEE Trans. Automat. Control*, Vol. 53, No. 10, pp.2402-2408, 2008.

[153] J.-X. Xu and D. Huang, "Initial State Iterative Learning for Final State Control in Motion Systems", *Automatica*, Vol. 44, pp.3162-3169, 2008.

[154] J.-X. Xu, D. Huang and S. Pindi, "Optimal Tuning of PID Parameters Using Iterative Learning Approach", *SICE Journal of Control, Measurement, and System Integration*, Vol. 1, No. 2, pp.143-154, 2008.

[155] J.-X. Xu, W. Wang and D. Huang, "Iterative Learning in Ballistic Control", Submitted.

[156] J.-X. Xu and D. Huang, "Discrete-Time Adaptive Control for Nonlinear Systems with Periodic Parameters: A Lifting Approach", Submitted.

[157] J.-X. Xu, W. Wang and D. Huang, "Iterative Learning in Ballistic Control", In *IEEE American Control Conf.*, July, New York, USA, 2007.

[158] J.-X. Xu and D. Huang, "Optimal Tuning of PID Parameters Using Iterative Learning Approach", In *IEEE Multiple Conf. on Systems and Control*, October, Singapore, 2007.

[159] J.-X. Xu and D. Huang, "Spatial Periodic Adaptive Control for Rotary Machine Systems", In *Int. Federation of Automatic Control (IFAC)*, July, Korea, 2008.

[160] J.-X. Xu and D. Huang, "Initial State Iterative Learning for Final State Control in Motion Systems", In *Int. Federation of Automatic Control (IFAC)*, July, Korea, 2008.

[161] J.-X. Xu and D. Huang, "Discrete-time adaptive control for nonlinear systems with periodic parameters: A lifting approach", In *7th Asian Control Conf.*, August, Hong Kong, 2009.

[162] C. Yang, S. S. Ge, C. Xiang, T. Chai, and T. H. Lee, "Output feedback NN control for two classes of discrete-time systems with unknown control directions in a unified approach", *IEEE Trans. on Neural Networks*, Vol. 19, No. 11, pp.1873-1886, 2008.

[163] C. Yang, S. S. Ge, and T. H. Lee, "Output feedback adaptive control of a class of nonlinear discrete-time systems with unknown control directions", *Automatica*, Vol. 45, pp.270-276, 2009.

[164] H. Yamaura, and K. Ono, "Vibrationless velocity control of a positioning mechanism for high-order modes of vibration", In *Proc. 2nd Int. Conf. Motion and Vibration Control*, Vol. 2, pp.605-610, 1994.

[165] Yasutaka Baba, et al, "Design of control method to rotate pendulum", In *SICE-ICASE, International Joint Conf.*, pp.2381-2385, Oct. 2006.

[166] P.-C. Yeh and P. V. Kokotović, "Adaptive control of a class of nonlinear discrete-time systems", *Int. J. Control*, Vol. 62, pp.303-324, 1995.

[167] X. Yue, D. M. Vilathgamuwa and K. J. Tseng, "Observer-based robust adaptive control of PMSM with initial rotor position uncertainty", *IEEE Trans. on Inductry Applications*, Vol. 39, No. 3, pp.645-656, 2003.

[168] A. T. zaremba, I. V. Burkov and R. M. Stuntz, "Active damping of engine speed oscillations based on learning control", In *Proc. 1998 American Control Conf.*, pp. 2143-2147, Philadelphia, PA, USA, June 24-26, 1998.

[169] Y. Zhang, C. Wen, and Y. C. Soh, "Robust adaptive control of uncertain discrete-time systems ", *Automatica*, Vol. 35, pp.321-329, 1999.

[170] J. Zhao and I. Kanellakopoulos, "Active identification for discrete-time nonlinear control-part I: output-feedback systems", *IEEE Trans. Automat. Control*, Vol. 47, No. 2, pp.210-224, 2002.

[171] J. Zhao and I. Kanellakopoulos, "Active identification for discrete-time nonlinear control-part II: strict-feedback systems", *IEEE Trans. Automat. Control*, Vol. 47, No. 2, pp.225-240, 2002.

[172] Y. Zhang, C. Wen, and Y. C. Soh, "Discrete-time robust backstepping adaptive control for nonlinear time-varying systems ", *IEEE Trans. Automat. Control*, Vol. 45, No. 9, pp.1749-1755, 2000.

[173] Y. Zhang, C. Wen, and Y. C. Soh, "Robust adaptive control of nonlinear discrete-time systems by backstepping without overparameterization", *Automatica*, Vol. 37, pp.551-558, 2001.

[174] Y. Zhang, W. H. Chen, and Y. C. Soh, "Improved robust backstepping adaptive control for nonlinear discrete-time systems without overparameterization", *Automatica*, Vol. 44, pp.864-867, 2008.

Appendix A: Algorithms and Proof Details

A.1: Proof of Proposition 2.3

First we apply the principle of induction to prove the relationship

$$z_{j+1} = \frac{N_j(x_1, x_2, \cdots, x_{j+1})}{x_1^{2j-1}}, \tag{8.1}$$

where N_j is a polynomial of $x_1, x_2, \cdots, x_{j+1}$.

When $j = 1$,

$$\nabla z_1 = \nabla x_1 = \frac{x_2}{x_1}.$$

From the state transformation (2.13),

$$z_2 = \nabla x_1 = \frac{N_1(x_1, x_2)}{x_1}$$

and $N_1 = x_2$.

When $j = 2$,

$$\begin{aligned} \nabla z_2 &= \nabla^2 x_1 = \nabla \left(\frac{N_1(x_1, x_2)}{x_1} \right) \\ &= \frac{x_1 \nabla N_1 - N_1 \nabla x_1}{x_1^2} \end{aligned} \tag{8.2}$$

Note that $\nabla x_j = x_j / x_1$,

$$\nabla N_1 = L_{[\nabla x_1,\, \nabla x_2]} N_1 = L_{[\frac{x_2}{x_1},\, \frac{x_3}{x_1}]} N_1 = \frac{1}{x_1} L_{[x_2,\, x_3]} N_1.$$

Substituting ∇N_1 into (8.2) yields

$$\begin{aligned} z_3 &= \nabla^2 x_1 = \frac{x_1 L_{[x_2,\, x_3]} N_1 - x_2 N_1}{x_1^3} = \frac{N_2}{x_1^3} \\ N_2 &\triangleq x_1 L_{[x_2,\, x_3]} N_1 - x_2 N_1. \end{aligned} \tag{8.3}$$

It can be seen that N_1 and N_2 are polynomials. The expressions of z_2 and z_3 are consistent with (8.1).

Now assume

$$z_j = \frac{N_{j-1}(x_1, x_2, \cdots, x_j)}{x_1^{2j-3}}. \tag{8.4}$$

Our objective is to prove (8.1).

Note that N_{j-1} is a function of the arguments x_1, x_2, $\cdots$, x_j, by differentiation we have

$$\begin{aligned} z_{j+1} &= \nabla^j x_1 = \nabla z_j = \nabla \left(\frac{N_{j-1}}{x_1^{2j-3}} \right) \\ &= \frac{x_1^{2j-3} \nabla N_{j-1} - N_{j-1} \nabla (x_1^{2j-3})}{x_1^{2(2j-3)}}. \end{aligned} \tag{8.5}$$

Analogous to the preceding derivation,

$$\nabla N_{j-1} = L_{[\nabla x_1, \cdots, \nabla x_j]} N_{j-1} = \frac{1}{x_1} L_{[x_2, \cdots, x_{j+1}]} N_{j-1},$$

as well as

$$\nabla (x_1^{2j-3}) = (2j-3) x_1^{2j-4} \nabla x_1 = (2j-3) x_1^{2j-5} x_2.$$

Substituting the above relations into (8.5) yields

$$\begin{aligned} z_{j+1} &= \frac{x_1^{2j-5} L_{[x_2, \cdots, x_{j+1}]} N_{j-1} - (2j-3) x_1^{2j-5} x_2 N_{j-1}}{x_1^{2(2j-3)}} \\ &= \frac{x_1 L_{[x_2, \cdots, x_{j+1}]} N_{j-1} - (2j-3) x_2 N_{j-1}}{x_1^{2j-1}} \end{aligned} \tag{8.6}$$

which is consistent with (8.1).

Next we derive the dynamics of ∇z_n

$$\nabla z_n = \nabla \left(\frac{N_{n-1}}{x_1^{2n-1}} \right). \tag{8.7}$$

The differentiation of N_{n-1} is

$$\nabla N_{n-1} = L_{[\nabla x_1, \cdots, \nabla x_n]} N_{j-1} = \frac{1}{x_1} L_{[x_2, \cdots, x_n]} N_{n-1} + \frac{\partial N_{n-1}}{\partial x_n} \nabla x_n$$

and

$$\nabla x_n = \mathbf{a}(s)^T \boldsymbol{\zeta}(\mathbf{x}) + b(s)u.$$

Substituting the above relations into (8.7) we obtain

$$\begin{aligned} \nabla z_n &= \frac{x_1 L_{[x_2, \cdots, x_n]} N_{n-1} - (2n-3) x_2 N_{n-1}}{x_1^{2n-1}} + \frac{1}{x_1^{2n-2}} \frac{\partial N_{n-1}}{\partial x_n} [\mathbf{a}(s)^T \boldsymbol{\zeta}(\mathbf{x}) + b(s)u] \\ &= \mathbf{a}(s)^T \boldsymbol{\xi}_x^0(\mathbf{x}) + \rho_x(\mathbf{x}) + b(s)\eta_x(\mathbf{x})u \end{aligned} \tag{8.8}$$

where

$$\eta_x = \frac{1}{x_1^{2n-2}} \frac{\partial N_{n-1}}{\partial x_n}, \qquad \boldsymbol{\xi}_x^0(\mathbf{x}) = \eta_x(\mathbf{x}) \boldsymbol{\zeta}^T(\mathbf{x});$$
$$\rho_x = \frac{x_1 L_{[x_2, \cdots, x_n]} N_{n-1} - (2n-3) x_2 N_{n-1}}{x_1^{2n-1}}.$$

Finally we prove the transformation $\mathbf{z} = \mathcal{T}(x_1, \cdots, x_n)$ is diffeomorphism, i.e., its inverse transformation exists and is smooth. Again we apply the principle of induction to prove a general relationship

$$\begin{aligned} x_{j+1} &= z_1^j z_{j+1} + \frac{f_{j+1}(z_1, \cdots, z_j)}{z_1^{l_{j+1}}}, \\ \frac{\partial N_j}{\partial x_{j+1}} &= x_1^{j-1} > 0, \end{aligned} \tag{8.9}$$

where f_{j+1} is a polynomial of $z_1, z_2, \cdots, z_j$, and l_{j+1} is a non-negative integer.

First, we have

$$x_1 = z_1, \qquad x_2 = z_1 z_2$$

and

$$\frac{\partial N_1}{\partial x_2} = 1$$

which are consistent with (8.9). Next assume

$$\begin{array}{rcl} x_i & = & z_1^{i-1} z_i + \frac{f_i(z_1,\cdots,z_{i-1})}{z_1^{l_i}}, \\ \frac{\partial N_{i-1}}{\partial x_i} & = & x_1^{i-2}, \end{array} \tag{8.10}$$

hold for $i = 1, \cdots, j$. From (8.6) and using the relationship (8.10)

$$\begin{array}{rcl} z_{j+1} & = & \dfrac{x_1 L_{[x_2,\cdots,x_j]} N_{j-1} - (2j-3) x_2 N_{j-1}}{x_1^{2j-1}} + \dfrac{\partial N_{j-1}}{\partial x_j} \dfrac{x_{j+1}}{x_1^{2j-2}} \\ & = & \dfrac{x_1 L_{[x_2,\cdots,x_j]} N_{j-1} - (2j-3) x_2 N_{j-1}}{x_1^{2j-1}} + \dfrac{x_{j+1}}{x_1^{j}}. \end{array} \tag{8.11}$$

Since $x_1 = z_1$, from (8.11) solving for x_{j+1} yields

$$x_{j+1} = z_1^j z_{j+1} + \frac{x_1 L_{[x_2,\cdots,x_j]} N_{j-1} - (2j-3) x_2 N_{j-1}}{x_1^{j-1}}. \tag{8.12}$$

The polynomial N_{j-1} consists of x_1, $\cdots$, x_j. By substituting x_i in the second term on the right hand side of (8.12) with (8.10),

$$\frac{x_1 L_{[x_2,\cdots,x_j]} N_{j-1} - (2j-3) x_2 N_{j-1}}{x_1^{j-1}}$$

becomes a function of z_1, $\cdots$, z_j, and the denominator consists of z_1 only. As a result, the relationship (8.9) holds.

In terms of (8.9), the relations between x_j and z_j can be summarized in a matrix form

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & z_1 & 0 & \cdots & 0 & 0 \\ \star & \star & z_1^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \star & \star & \star & \cdots & z_1^{n-2} & 0 \\ \star & \star & \star & \cdots & \star & z_1^{n-1} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}, \tag{8.13}$$

where elements denoted by $\star$ at the jth row are fractions of $\frac{f_j(z_1,\cdots,z_{j-1})}{z_1^{l_j}}$ and therefore continuous and nonsingular when $x_1 > 0$. (8.13) shows that the inverse transformation $\mathbf{x} = \mathcal{T}^{-1}(\mathbf{z})$ exists and is smooth.

A.2: The procedure of future states prediction in Chapter 3

Let $\hat{\boldsymbol{\theta}}_{l,k}$ and $\hat{\mathbf{b}}_{l,k}$ denote the estimates of $\boldsymbol{\theta}_l$ and $\mathbf{b}_l$ at the k-th step, respectively. For convenience, we denote $\hat{\vartheta}_{l,k} = \left[\hat{\boldsymbol{\theta}}_{l,k}^T, \hat{\mathbf{b}}_{l,k}^T\right]^T$. Define one-step prediction $\hat{x}_l(k+1|k)$, the estimate of $x_{l,k+1}$ is

$$\hat{x}_l(k+1|k) = \hat{\vartheta}_{l,k-n+2}^T \psi_{l,k}, \quad l = 1, 2, \cdots, n-1 \tag{8.14}$$

where $\psi_{l,k} = \left[\bar{\boldsymbol{\xi}}_{l,k}^T, \bar{\mathbf{x}}_{l+1,k}^T\right]^T$. Define two-step prediction $\hat{x}_l(k+2|k)$, the estimate of $x_{l,k+2}$ is

$$\hat{x}_l(k+2|k) = \hat{\vartheta}_{l,k-n+3}^T \hat{\psi}_l(k+1|k), \tag{8.15}$$

where $l = 1, 2, \cdots, n-2$, and

$$\begin{aligned}
\hat{\psi}_l(k+1|k) &= \left[\bar{\boldsymbol{\xi}}_l(\hat{\mathbf{x}}_l(k+1|k))^T, \bar{\mathbf{x}}_{l+1}(k+1|k)^T\right]^T, \\
\hat{\mathbf{x}}_l(k+1|k) &= [\hat{x}_1(k+1|k), \hat{x}_2(k+1|k), \\
&\qquad \cdots, \hat{x}_l(k+1|k)]^T, \\
\bar{\mathbf{x}}_{l+1}(k+1|k) &= \boldsymbol{\zeta}_{l,k+1} \hat{x}_{l+1}(k+1|k).
\end{aligned} \tag{8.16}$$

Define j-step ($j = 3, 4, \cdots, n-1$) prediction $\hat{x}_l(k+j|k)$, the estimate of $x_{l,k+j}$ is

$$\hat{x}_l(k+j|k) = \hat{\vartheta}_{l,k-n+j+1}^T \hat{\psi}_l(k+j-1|k), \tag{8.17}$$

where $l = 1, 2, \cdots, n-j$, and

$$\hat{\psi}_l(k+j-1|k) = \left[\bar{\boldsymbol{\xi}}_l(\hat{\mathbf{x}}_l(k+j-1|k))^T, \bar{\mathbf{x}}_{l+1}(k+j-1|k)^T\right]^T, \hat{\mathbf{x}}_l(k+j-1|k)$$

$$= [\hat{x}_1(k+j-1|k), \cdots, \hat{x}_l(k+j-1|k)]^T, \bar{\mathbf{x}}_{l+1}(k+j-1|k)$$

$$= \boldsymbol{\zeta}_{l,k+j-1}\hat{x}_{l+1}(k+j-1|k). \tag{8.18}$$

The parameter estimates in state prediction are calculated from the following updating law

$$\begin{aligned} \hat{\vartheta}_{l,k+1} &= \hat{\vartheta}_{l,k-n+2} - \frac{\tilde{x}_l(k+1|k)\psi_{l,k}}{1+\psi_{l,k}^T\psi_{l,k}}, \\ \tilde{x}_l(k+1|k) &= \hat{x}_l(k+1|k) - x_{l,k+1}, \\ \hat{\vartheta}_{l,k} &= \left[\hat{\boldsymbol{\theta}}_{l,k}^T, \hat{\mathbf{b}}_{l,k}^T\right]^T, \quad l = 1, 2, \cdots, n-1. \end{aligned} \tag{8.19}$$

Thus, we can predict Ψ_{k+n-1} as the following

$$\begin{aligned} \hat{\Psi}(k+n-1|k) &= \big[\Psi_1^T(\hat{\mathbf{x}}_1(k+n-1|k)), \\ &\qquad \Psi_2^T(\hat{\mathbf{x}}_2(k+n-2|k)), \cdots, \Psi_n^T(\mathbf{x}_n(k))\big]^T \end{aligned} \tag{8.20}$$

where each $\Psi_i^T(\hat{\mathbf{x}}_i(k+n-i|k))$ is the prediction of $\Psi_i^T(\mathbf{x}_i(k+n-i))$.

A.3: Parallel parametric adaptation laws in Chapter 3

Assuming $k = \kappa_b s + i, i = 0, 1, \cdots, \kappa_b - 1$, the parameter estimates in the control law are updated with respect to s by the following update law:

$$\epsilon_{i,s} = \frac{\gamma e_{\kappa_b(s-n)+n+i} + N_{i,s}\phi_{i,s}\beta_{i,s}}{G_{i,s}}, \quad \gamma > 0, \quad s \geq 0, \tag{8.21}$$

$$\begin{aligned} \hat{\Theta}_g(\kappa_b s + i) &= \hat{\Theta}_g(\kappa_b(s-n)+i) \\ &\quad + \gamma\frac{N_{i,s}}{D_{i,s}}\bar{\Psi}_{\kappa_b(s-n)+n+i-1}\epsilon_{i,s}, \end{aligned} \tag{8.22}$$

$$\begin{aligned} \hat{\mathbf{g}}_0(\kappa_b s + i) &= \hat{\mathbf{g}}_0(\kappa_b(s-n)+i) \\ &\quad - \gamma\frac{N_{i,s}}{D_{i,s}}\mathbf{r}_{\kappa_b(s-n)+n+i}\epsilon_{i,s}, \end{aligned} \tag{8.23}$$

$$\hat{\Theta}_g(j) = \mathbf{0}_{\kappa\kappa_b}, \hat{\mathbf{g}}_0(j) = \mathbf{0}_{\kappa_b}, j = 0, -1, \cdots, -\kappa_b n + 1,$$

$$
\begin{aligned}
\phi_{i,s} &\triangleq \phi(\kappa_b(s-n)+n+i), \\
z_{i,s} &\triangleq z(\kappa_b(s-n)+n+i), \\
\Delta\phi_{i,s} &= \phi_{i,s+1}-\phi_{i,s} = \frac{-N_{i,s}\beta_{i,s}\epsilon_{i,s}}{D_{i,s}}, \quad \phi_{i,-1}=0, \qquad (8.24)\\
\Delta z_{i,s} &= z_{i,s+1}-z_{i,s} = \frac{G_{i,s}\epsilon_{i,s}^2}{D_{i,s}}, \quad z_{i,-1}=0, \qquad (8.25)\\
\beta_{i,s} &= \beta(\kappa_b(s-n)+n+i-1) \qquad (8.26)\\
&= \hat{\Theta}_g^T(\kappa_b(s-n)+i)\times \\
&\quad \bar{\bar{\Psi}}(\kappa_b(s-n)+n+i-1|\kappa_b(s-n)+i), \\
N_{i,s} &= N(\chi_{i,s}), \quad \chi_{i,s} = z_{i,s}+\frac{\phi_{i,s}^2}{2}, \qquad (8.27)\\
G_{i,s} &= 1+|N_{i,s}|, \qquad (8.28)\\
D_{i,s} &= (1+|\phi_{i,s}|)(1+N_{i,s}^3) \\
&\quad \times \left(1+\|\bar{\Psi}_{\kappa_b(s-n)+n+i-1}\|^2\right. \\
&\quad \left. +\|\mathbf{r}_{\kappa_b(s-n)+n+i}\|^2+\beta_{i,s}^2+\epsilon_{i,s}^2\right),
\end{aligned}
$$

where $\epsilon_{i,s}$ are introduced as augmented errors, $\gamma>0$ is the tuning parameter, and $N(\chi_{i,s})$ is the discrete Nussbaum gain defined to be

$$
N(\chi_{i,s}) = \bar{\chi}_{i,s}\varpi_N(\chi_{i,s}), \quad \bar{\chi}_{i,s} = \sup_{s'\le s}\{\chi_{i,s'}\}
$$

and $\varpi_N(\chi_{i,s})$ is the sign function of the discrete Nussbaum gain, i.e., $\varpi_N(\chi_{i,s})=\pm 1$.

A.4: Proof of Property 4.1

(1) Initial position tuning for final position control

Look into the phase plane in Fig.8.1, two solution trajectories $\widehat{AB}$ and $\widehat{CD}$ represent solution trajectories of the dynamics (4.2) with different initial positions $u_x^* < u_x$. By

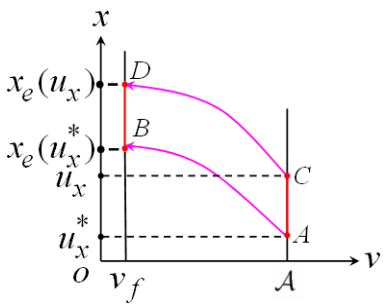


Figure 8.1: Initial position tuning for final position control.

virtue of the uniqueness of the solution, two trajectories do not intersect each other. As a result, $x(v, u_x^*) < x(v, u_x)$ and so is $x_e(u_x^*) < x_e(u_x)$. Therefore we have $(u_x - u_x^*)[x_e(u_x) - x_e(u_x^*)] > 0$.

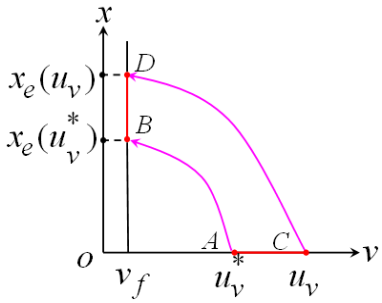


Figure 8.2: Initial speed tuning for final position control.

(2) Initial speed tuning for final position control

In Fig.8.2, the trajectory $\widehat{AB}$ starts from the initial speed u_v^* and the trajectory $\widehat{CD}$ starts from the initial speed u_v, while the initial displacements are zero. From Fig.8.2 and the uniqueness of solution, $u_v > u_v^*$ leads to the positions $x_e(u_v) > x_e(u_v^*)$ at the points D and B corresponding to the prespecified speed v_f. As a result we have $(u_v - u_v^*)[x_e(u_v) - x_e(u_v^*)] > 0$.

(3) Initial position tuning for final speed control

When $u_x > u_x^*$, from phase plane Fig. 8.3 we can see that the trajectory $\widehat{CD}$ is above the trajectory $\widehat{AB}$ because of the uniqueness of solution. When both positions drop to

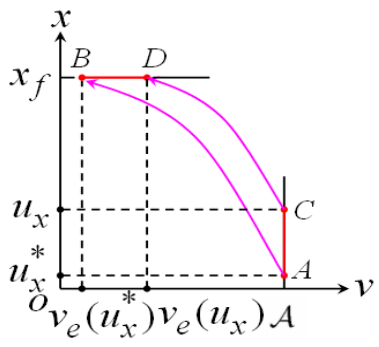


Figure 8.3: Initial position tuning for final speed control.

the same level at x_f, the speed D is obviously farther than the speed B. Therefore we have $(u_x - u_x^*)[v_e(u_x) - v_e(u_x^*)] > 0$.

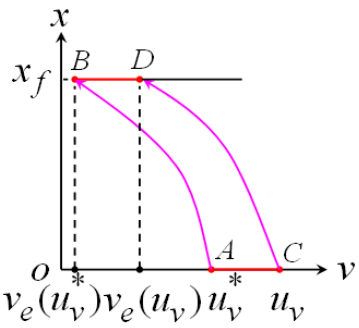


Figure 8.4: Initial speed tuning for final speed control.

(4) Initial speed tuning for final speed control

From Fig.8.4 and the uniqueness of solution, we can see that trajectory $\widehat{AB}$ with initial speed u_v^* is always on the left of the trajectory $\widehat{CD}$ with the initial speed u_v, because $u_v > u_v^*$. Accordingly $v_e(u_v) > v_e(u_v^*)$, that is, the point D is on the right of the point B. As a result we have $(u_v - u_v^*)[v_e(u_v) - v_e(u_v^*)] > 0$.

A.5: Proof of Lemma 4.1

Since $|z_d - z_i| \leq \lambda |u_d - u_i|$, there exists a quantity $0 < \lambda_i \leq \lambda$ such that

$$|z_d - z_i| = \lambda_i |u_d - u_i|. \tag{8.29}$$

Let $\gamma = r/\lambda$, from the constraint of γ we have $1-\rho < r < 1+\rho$. Substituting (8.106) into (4.8) yields

$$|u_d - u_{i+1}| = |1-\gamma\lambda_i||u_d - u_i| = |1 - r\frac{\lambda_i}{\lambda}||u_d - u_i|.$$

The convergence of iteration learning is determined by the magnitude of the factor $|1 - r\frac{\lambda_i}{\lambda}|$. The upper bound for $|1 - r\frac{\lambda_i}{\lambda}|$ indicates the slowest convergence rate. Next we derive this upper bound with two cases.

Case 1. $\min\{\frac{\lambda}{\lambda_i}, 1+\rho\} = \frac{\lambda}{\lambda_i}$. When $1-\rho < r \leq \frac{\lambda}{\lambda_i}$,

$$|1 - r\frac{\lambda_i}{\lambda}| = 1 - r\frac{\lambda_i}{\lambda} < 1 - (1-\rho)\frac{\lambda_i}{\lambda} \stackrel{\triangle}{=} \rho_i < 1.$$

When $\frac{\lambda}{\lambda_i} < r < 1+\rho$,

$$\begin{aligned} |1 - r\frac{\lambda_i}{\lambda}| &= r\frac{\lambda_i}{\lambda} - 1 < (1+\rho)\frac{\lambda_i}{\lambda} - 1 \\ &\leq \rho = 1 - (1-\rho) \leq \rho_i. \end{aligned}$$

From $(1+\rho)\frac{\lambda_i}{\lambda} \leq 1+\rho$ we conclude $(1+\rho)\frac{\lambda_i}{\lambda} - 1 \leq \rho$ and, thus,

$$(1+\rho)\frac{\lambda_i}{\lambda} - 1 \leq \rho = 1 - (1-\rho) \leq \rho_i.$$

Case 2. $\min\{\frac{\lambda}{\lambda_i}, 1+\rho\} = 1+\rho$. In this case, we have

$$|1 - r\frac{\lambda_i}{\lambda}| = 1 - r\frac{\lambda_i}{\lambda} < 1 - (1-\rho)\frac{\lambda_i}{\lambda} = \rho_i.$$

Thus the upper bound of the convergence factor is

$$\rho_i = 1 - (1-\rho)\frac{\lambda_i}{\lambda}. \tag{8.30}$$

for all iterations. Note that when $u_i \neq u_d$, $z_i \neq z_d$ by the uniqueness of solution, consequently $\lambda_i \neq 0$ by (8.106) and the upper bound ρ_i will be strictly less than 1 as far as u_i does not converge to u_d.

Let ε denote the desired ε-precision bound of learning, i.e. $|z_d - z_i| < \varepsilon$. Now we show that the sequence z_i can enter the prespecified ε-precision bound after a finite number of iterations. Let M denote the initial input error $|u_d - u_x| = M < \infty$.

First, considering the fact $\rho_i \leq 1$, using (8.107) repeatedly yields

$$|z_d - z_i| = \lambda_i |u_d - u_i| = \lambda_i \prod_{j=1}^{i-1} \rho_j |u_d - u_x| \leq \lambda_i M.$$

Before z_i enters the ε-bound,

$$\varepsilon < |z_d - z_i| \leq \lambda_i |u_d - u_x| \leq \lambda_i M$$

which gives the lower bound of the coefficient λ_i, $\lambda_i \geq \epsilon/M$ for all iterations before learning terminates. Similarly by using the relationship (8.107) repeatedly, and substituting the lower bound of λ_i, we can derive

$$\begin{aligned} |z_d - z_i| &\leq \lambda |u_d - u_i| \leq \lambda \prod_{j=1}^{i-1} \rho_j |u_d - u_x| \\ &= \lambda \prod_{j=1}^{i-1} \left(1 - (1-\rho)\frac{\lambda_j}{\lambda}\right) M \leq M\lambda \left(1 - (1-\rho)\frac{\epsilon}{M\lambda}\right)^i \end{aligned}$$

which gives the upper bound of $|z_d - z_i|$. Solving for $M\lambda \left(1 - (1-\rho)\frac{\epsilon}{M\lambda}\right)^{i-1} \leq \varepsilon$ with respect to i, the maximum number of iterations needed is

$$i \leq \frac{\log \frac{\varepsilon}{M\lambda}}{\log\left(1 - (1-\rho)\frac{\varepsilon}{M\lambda}\right)} + 1.$$

A.6: Proof of Theorem 4.1

For simplicity, in subsequent graphics we demonstrate $u_{x,i} > u_{x,d}$ or $u_{v,i} > u_{v,d}$ only. By following the same derivation procedure, we can easily prove learning convergence for opposite cases $u_{x,i} < u_{x,d}$ or $u_{v,i} < u_{v,d}$. Denote $\widehat{AB}$ the trajectories of (4.2) associated with the desired control inputs, and $\widehat{CD}$ the trajectories associated with the actual control inputs at the ith iteration.

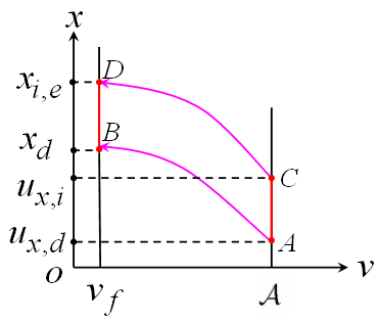


Figure 8.5: Phase portrait of system (4.2) in v-x plane with initial position learning for final position control.

(i) Initial position iterative learning for final position control

The initial speed is fixed at $\mathcal{A}$. Denote $u_{x,d}$ the desired initial position that achieves the desired final position x_d at the prespecified speed v_f, that is, applying $u_{x,d}$ to the dynamics (4.2) yields $x_e = x_d$.

Integrating (4.2) yields

$$\begin{aligned} & x_d - x_{i,e} \\ = \; & u_{x,d} - u_{x,i} - \int_{\mathcal{A}}^{v_f} \left[g\left(v, x(v, u_{x,d})\right) - g\left(v, x(v, u_{x,i})\right) \right] dv. \end{aligned}$$

Applying the Lipschitz continuity condition (4.3) yields

$$\begin{aligned} & |x_d - x_{i,e}| \\ \leq \; & |u_{x,d} - u_{x,i}| + \int_{v_f}^{\mathcal{A}} L(v) |x(v, u_{x,d}) - x(v, u_{x,i})| dv. \end{aligned} \tag{8.31}$$

Define $\lambda = \exp\left(\int_{v_f}^{\mathcal{A}} L(v) dv\right)$. Applying the generalized Grownwall inequality to (8.31) we obtain $|x_d - x_{i,e}| \leq \lambda |u_{x,d} - u_{x,i}|$. As shown in Fig. 8.5, $\overline{BD} = |x_d - x_{i,e}| \leq \lambda |u_{x,d} - u_{x,i}| = \lambda \overline{AC}$. Therefore, choose a $\rho < 1$ and the learning gain according to λ and (4.9), the learning convergence is obtained.

(ii) Initial speed iterative learning for final position control

As shown in Fig.8.6, draw a line $\overline{AE}$ starting from A such that it parallels the x-axis,

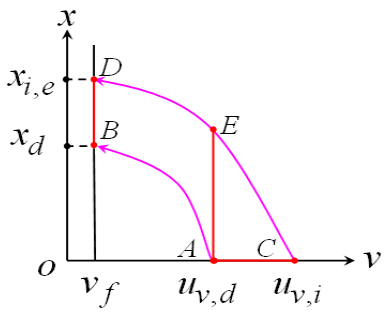


Figure 8.6: Phase portrait of system (4.2) in v-x plane with initial speed learning for final position control.

where E is the point intersected with $\widehat{CD}$. In order to find the relationship between the initial speed and final position, we first derive the relationship between $\overline{BD}$ and $\overline{AE}$, then derive the relationship between $\overline{AE}$ and $\overline{AC}$.

Using the result of case (i), we can obtain the relationship between the initial position difference $\overline{AE}$ and final position difference $\overline{BD}$

$$\overline{BD} \leq \lambda_1 \overline{AE}, \qquad \lambda_1 = \exp\left(\int_{v_f}^{\mathcal{A}} L(v)dv\right). \tag{8.32}$$

Next investigate the relationship between the position difference $\overline{AE}$ and initial speed difference $\overline{AC}$. Denote x^* the position at E. Integrating (4.2), the position difference $\overline{AE}$ at the ith iteration can be estimated using the mean value theorem

$$\begin{aligned}
\overline{AE} &= x^* - 0 = -\int_{u_{v,i}}^{u_{v,d}} g\left(v, x(v, u_{v,i})\right) dv \\
&= g\left(v, x(v, u_{v,i})\right)\left(u_{v,i} - u_{v,d}\right) \qquad \exists v \in [u_{v,d}, u_{v,i}] \\
&\leq \max_{v\in[u_{v,d},u_{v,i}]} g\left(v, x(v, u_{v,i})\right) \cdot |u_{v,i} - u_{v,d}|.
\end{aligned} \tag{8.33}$$

Using this bounding condition $g(v, x) \leq g_1(v)$ and (8.33), we obtain

$$\begin{aligned}
&\max_{v\in[u_{v,d},u_{v,i}]} g\left(v, x(v, u_{v,i})\right) \\
\leq &\max_{v\in[u_{v,d},u_{v,i}]} g_1(v) \leq \max_{v\in[v_f,\mathcal{A}]} g_1(v).
\end{aligned}$$

Define $\lambda_2 = \max_{v\in[v_f,\mathcal{A}]} g_1(v)$, we have $\overline{AE} \leq \lambda_2\overline{AC}$, $\overline{BD} \leq \lambda_1\overline{AE} \leq \lambda\overline{AC}$ where $\lambda = \lambda_1\lambda_2$. Therefore, choose $\rho < 1$ and the learning gain according to λ and (4.9), the learning convergence is guaranteed.

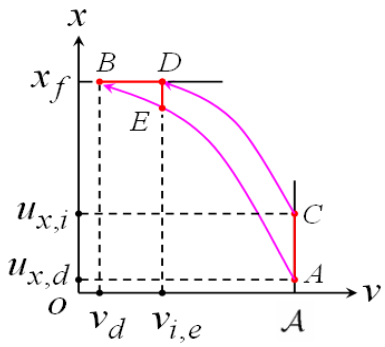


Figure 8.7: Phase portrait of system (4.2) in v-x plane with initial position learning for final speed control.

(iii) Initial position iterative learning for final speed control

As shown in Fig.8.7, draw a line through point D such that it parallels the x-axis and intersects the trajectory $\widehat{AB}$ at the point E. Denote x^* the position at E. In order to find the relationship between the initial position and final speed, we first derive the relationship between $\overline{AC}$ and $\overline{ED}$, then derive the relationship between $\overline{ED}$ and $\overline{BD}$.

Using the result of case (i), we can obtain the relationship between the initial position difference $\overline{AC}$ and final position difference $\overline{ED}$

$$\overline{ED} \leq \lambda_1\overline{AC}, \qquad \lambda_1 = \exp\left(\int_{v_d}^{\mathcal{A}} L(v)dv\right). \tag{8.34}$$

Next investigate the relationship between the initial position difference $\overline{ED}$ and the final speed difference $\overline{BD}$. Integrating (4.2), the speed difference $\overline{ED}$ at the ith iteration can be estimated using the mean value theorem

$$\begin{aligned}\overline{ED} &= x_f - x^* = \int_{v_d}^{v_{i,e}} g\left(v, x(v, x^*)\right) dv \\ &= g\left(v, x(v, x^*)\right)(v_{i,e} - v_d) \qquad \exists v \in [v_d, v_{i,e}]\end{aligned}$$

$$\geq \min_{v\in[v_d,v_{i,e}]} g\left(v, x(v,x^*)\right)(v_{i,e}-v_d). \tag{8.35}$$

Substitute the relationship $\min_{v\in[v_d,v_{i,e}]} g\left(v, x(v,x^*)\right) \geq c$ into (8.35) and note $\overline{BD} = v_{i,e} - v_d$, we have $\overline{BD} \leq \lambda_2 \overline{ED}, \lambda_2 = 1/c$. Finally using (8.34) it can be derived that $\overline{BD} \leq \lambda_2 \overline{ED} \leq \lambda \overline{AC}$ where $\lambda = \lambda_1\lambda_2$. Therefore, choose $\rho < 1$ and the learning gain according to λ and (4.9), the learning convergence is guaranteed.

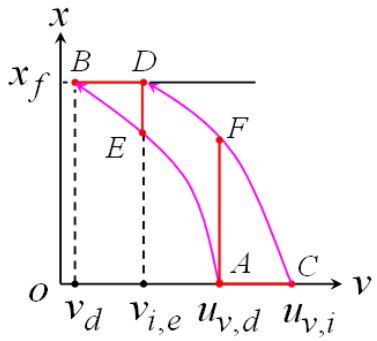


Figure 8.8: Phase portraying of system (4.2) in v-x plane with initial speed learning for final speed control.

(iv) Initial speed iterative learning for final speed control

As shown in Fig.8.8, the learning convergence in this case can be derived directly by using the results of cases (i), (ii) and (iii). Draw two straight lines $\overline{ED}$ and $\overline{AF}$. There exist three relations. The first relationship is between the initial speed difference $\overline{AC}$ and the final position difference $\overline{AF}$, which has been discussed in the second part of case (ii). The second relationship is between the initial position difference $\overline{AF}$ and the final position difference $\overline{ED}$, which has been explored in case (i). The third relationship is between the initial position difference $\overline{ED}$ and the final speed difference $\overline{BD}$, which was given in the second part of case (iii). Therefore the value of λ given in the theorem consists of three factors

$$\max_{v\in[v_f,\mathcal{A}]} g_1(v), \qquad \exp\left(\int_{v_f}^{\mathcal{A}} L(v)dv\right), \qquad \frac{1}{c}.$$

A.7: Proof of Theorem 4.3

The existence and monotonicity of solution $v(x,k)$ with respect to k in system (4.13) imply the existence and uniqueness of desired parameter k_d. With similar discussion as in *Property* 4.1, we can see that the positivity (negativity) of difference $k_d - k_i$ always induces the non-positivity (non-negativity) of $v_d - v_{i,e}$. Thus, it is sufficient to give the ϑ-relationship between $|v_d - v_{i,e}|$ and $|k_d - k_i|$.

(i) $\lim_{t\to\infty} x(t,k_i) > x_f$. Integrating system (4.13) in the interval $[0, x_f]$ and using *Assumption* 4.3 yield

$$|v_d - v_{i,e}| \leq \int_0^{x_f} L_1 |v(x,k_d) - v(x,k_i)| dx + |k_d - k_i| \int_0^{x_f} L_2 dx,$$

which further implies $|v_d - v_{i,e}| \leq \lambda |k_d - k_i|$ by the generalized Grownwall Lemma.

(ii) $\lim_{t\to\infty} x(t,k_i) \leq x_f$. In this case, $v_{i,e} = 0$. Let k_* be the unique parameter input determined by

$$0 = v_* = v(x_f, k_*) = \mathcal{A} - \int_0^{x_f} g(x, v(x,k_*), k_*) dx.$$

By the monotonic decreasing property of $v_{i,e}$ with respect to k_i, the relationship $v_{i,e} = v_* = 0 < v_d$ implies that $k_d < k_* \leq k_i$. Thus,

$$|v_d - v_{i,e}| = |v_d - v_*| \leq \lambda |k_d - k_*| \leq \lambda |k_d - k_i|. \tag{8.36}$$

A.8: Proof of Theorem 5.1

Define the following time weighted composite energy function (CEF)

$$E_i(t) = \frac{1}{2} e^{-\lambda t} e_i^2(t) + \frac{1}{2q} \int_0^t e^{-\lambda \tau} \left(u_i - u_r\right)^2 d\tau \tag{8.37}$$

where $\lambda > 2L_\eta$ is a finite positive constant. Next, we prove that the CEF (8.37) is non-increasing along the iteration domain.

Let $\Delta E_i(t) \stackrel{\triangle}{=} E_i(t) - E_{i-1}(t)$ and $\Delta_h(u_i) \stackrel{\triangle}{=} u_i - u_r$. Then, a simple computation yields

$$\begin{aligned}\Delta E_i &= \frac{1}{2}e^{-\lambda t}\left[e_i^2(t) - e_{i-1}^2(t)\right] \\ &\quad + \frac{1}{2q}\int_0^t e^{-\lambda\tau}\left[\Delta_h^2(u_i) - \Delta_h^2(u_{i-1})\right]d\tau. \end{aligned} \tag{8.38}$$

We can compute each term on the right hand side of (8.38) separately. First, noting the error dynamics (5.11), it follows that

$$\begin{aligned}\frac{1}{2}e^{-\lambda t}e_i^2(t) &= -\frac{\lambda}{2}\int_0^t e^{-\lambda\tau}e_i^2(\tau)d\tau \\ &\quad + \int_0^t e^{-\lambda\tau}e_i(\tau)\left[\eta(x^r,\tau) + u_r(\tau)\right]d\tau \\ &\quad - \int_0^t e^{-\lambda\tau}e_i(\tau)\left[\eta(x_i,\tau) + u_i\right]d\tau \\ &\leq -\frac{\lambda - 2L_\eta}{2}\int_0^t e^{-\lambda\tau}(e_i^2(\tau))d\tau \\ &\quad + \int_0^t e^{-\lambda\tau}e_i(\tau)\left[u_r - u_i\right]d\tau. \end{aligned} \tag{8.39}$$

Now looking into the third term on the right hand side of (8.38). Using updating law (5.12) leads to

$$\begin{aligned}&\frac{1}{2q}\int_0^t e^{-\lambda\tau}\left[\Delta_h^2(u_i) - \Delta_h^2(u_{i-1})\right] \\ &= \frac{1}{2}\int_0^t e^{-\lambda\tau}e_i(\tau)\left[2u_i - qe_i - 2u_r\right] \\ &\leq \frac{1}{2}\int_0^t e^{-\lambda\tau}e_i(\tau)\left[2u_i - 2u_r\right] \\ &= \int_0^t e^{-\lambda\tau}e_i(\tau)\left[u_i - u_r\right]. \end{aligned} \tag{8.40}$$

Substituting (8.39) and (8.40) into (8.38),

$$\begin{aligned}\Delta E_i &\leq -\frac{\lambda - 2L_\eta}{2}\int_0^t e^{-\lambda\tau}e_i^2(\tau)d\tau - \frac{1}{2}e^{-\lambda t}e_{i-1}^2(t) \\ &\leq -\frac{1}{2}e^{-\lambda t}e_{i-1}^2(t) \leq 0, \end{aligned} \tag{8.41}$$

which shows that the energy function E_i is non-increasing along the iteration axis. The proof is completed by following the similar steps in [143, Theorem 1].

A.9: Proof of Lemma 5.2

Write (5.14) with a more concise form,

$$\begin{aligned}\dot{z} &= \dot{v}\left[D^{-1}\left(A-\beta\mathcal{S}(\dot{v})|z|^{n-1}z-\gamma|z|^n\right)\right]\\ &= \dot{v}\left[D^{-1}\left(A-|z|^n(\gamma+\beta\mathcal{S}(\dot{v}z))\right)\right].\end{aligned} \tag{8.42}$$

Thus, in each monotonic branch of $v(t)$

$$\frac{\partial z}{\partial v} = D^{-1}\left(A-|z|^n(\gamma+\beta\mathcal{S}\left(\dot{v}z\right))\right). \tag{8.43}$$

Next, consider the input-output monotonicity of hysteresis in four sub-cases.

Case (C_1). Applying the facts $|z(t)|\le z_a$ and $\gamma+\beta>0$ in this case, no matter what is the sign of $\dot{v}z$, we have

$$\begin{aligned}\frac{\partial z}{\partial v} &\ge D^{-1}\left(A-|z|^n(\gamma+\beta)\right)\\ &\ge D^{-1}\left(A-z_a^n(\gamma+\beta)\right)=0.\end{aligned} \tag{8.44}$$

On the other hand, since $\gamma-\beta\le 0$ in (C_1), it can be seen that

$$\begin{aligned}\frac{\partial z}{\partial v} &\le D^{-1}\left(A-|z|^n(\gamma-\beta)\right)\\ &= D^{-1}\left(A+|z|^n(\beta-\gamma)\right)\\ &\le D^{-1}\left(A+\frac{A}{\beta+\gamma}(\beta-\gamma)\right)\le 2AD^{-1}.\end{aligned} \tag{8.45}$$

Case (C_2). In this case, $\gamma-\beta>0$, $\beta>0$ and $|z|\le z_a$. Subsequently, $\gamma+\beta=\gamma-\beta+2\beta>0$. Thus, the non-negative property (8.44) in (C_1) still holds here. Moreover,

$$\frac{\partial z}{\partial v}\le D^{-1}\left(A-|z|^n(\gamma-\beta)\right)\le AD^{-1}. \tag{8.46}$$

Case (C_3'). Noticing that $|z(t)|\le z_b, \gamma+\beta>0$ and $\frac{\alpha}{1-\alpha}+\frac{2\beta A}{\beta-\gamma}\ge\frac{\epsilon}{(1-\alpha)k}$ in this case, we can similarly see that

$$\frac{\partial z}{\partial v} \ge D^{-1}\left(A-|z|^n(\gamma+\beta)\right)$$

$$\begin{aligned} &\geq D^{-1}\left(A - z_b^n(\gamma+\beta)\right) \\ &= \frac{2\beta A}{D(\beta-\gamma)} \geq -\frac{\alpha}{D(1-\alpha)} + \frac{\epsilon}{(1-\alpha)Dk}. \end{aligned} \tag{8.47}$$

On the other hand, owing to $\beta > 0$,

$$\begin{aligned} \frac{\partial z}{\partial v} &\leq D^{-1}\left(A - |z|^n(\gamma-\beta)\right) \\ &\leq D^{-1}\left(A - z_b^n(\gamma-\beta)\right) \\ &= D^{-1}\left(A - \frac{A}{\gamma-\beta}(\gamma-\beta)\right) = 0. \end{aligned} \tag{8.48}$$

Case (C_4'). Note that $\beta > 0$ and $|z(t)| \leq z_b$ in this case. Thus, the relationship (8.48) is still valid here. Moreover, considering $\beta + \gamma < 0,\ \beta \geq 0,$ and $\frac{\alpha}{1-\alpha} + A \geq \frac{\epsilon}{(1-\alpha)k}$,

$$\begin{aligned} \frac{\partial z}{\partial v} &\geq D^{-1}\left(A - |z|^n(\gamma+\beta)\right) \\ &\geq D^{-1}A \\ &\geq -\frac{\alpha}{D(1-\alpha)} + \frac{\epsilon}{(1-\alpha)Dk}. \end{aligned} \tag{8.49}$$

The bounds information for dz/dv is beneficial for us to estimate the bound of the input-output gradient in each branch of hysteresis. From (5.10), it follows that

$$\frac{\partial u}{\partial v} = \alpha k + (1-\alpha)Dk\frac{\partial z}{\partial v}.$$

In the cases (C_1) and (C_2), by using the inequalities (8.44)-(8.46)

$$\alpha k \leq \frac{\partial u}{\partial v} \leq \alpha k + 2(1-\alpha)kA \leq k\max\{1, 2A\}. \tag{8.50}$$

In the cases (C_3') and (C_4'), (8.47)-(8.49) imply that

$$\begin{aligned} \epsilon &= \alpha k + (1-\alpha)Dk\left(-\frac{\alpha}{D(1-\alpha)} + \frac{\epsilon}{(1-\alpha)Dk}\right) \\ &\leq \frac{\partial u}{\partial v} \leq k\alpha. \end{aligned} \tag{8.51}$$

Thus, combining (8.50) and (8.51) yields that

$$\min\{\epsilon, \alpha k\} \leq \frac{\partial u}{\partial v} \leq k\max\{1, \alpha, 2A\},$$

which is just (5.28).

A.10: Proof of Lemma 5.3

In the initial iteration, we have $\mathcal{S}(\dot{v}_0(t)) = \mathcal{S}(\dot{u}_{i,r}(t))$ by assumption. On the other hand, $\mathcal{S}(\dot{v}_0(t)) = \mathcal{S}(\dot{u}_0(t))$ and $\mathcal{S}(\dot{v}_{i,r}(t)) = \mathcal{S}(\dot{u}_{i,r}(t))$ by the input-output monotonicity property in each branch. Hence, the statement holds for $j = 0$. For any $j \geq 1$, assume $\mathcal{S}(\dot{v}_j(t)) = \mathcal{S}(\dot{u}_j(t)) = \mathcal{S}(\dot{v}_{i,r}(t)) = \mathcal{S}(\dot{u}_{i,r}(t))$. We further prove the statement holds in the $(j+1)$-th iteration. Specifically, differentiating the ILC law (7.4) in each monotonic branch of the j-th iteration,

$$\dot{v}_{j+1} = \dot{}_j v + q_h \Delta \dot{u}_j = \dot{}_j v - q_h \dot{u}_j + q_h \dot{u}_{i,r}. \tag{8.52}$$

If $\dot{v}_j(t) = 0$, then $\dot{u}_j(t) = 0$, implying that $\dot{v}_{j+1} = q_h \dot{u}_{i,r}$. Thus, $\mathcal{S}(\dot{v}_{j+1}) = \mathcal{S}(\dot{u}_{i,r}(t))$. If $\dot{v}_j(t) \neq 0$,

$$\begin{aligned} \dot{v}_{j+1} &= \dot{v}_j \left(1 - q_h \frac{\dot{u}_j}{\dot{v}_j}\right) + q_h \dot{u}_{i,r} \\ &= \dot{v}_j \left(1 - q_h \frac{du_j}{dv_j}\right) + q_h \dot{u}_{i,r} \end{aligned} \tag{8.53}$$

According to (5.28), we have $\epsilon \leq du_j/dv_j \leq \lambda$, implying

$$\begin{aligned} 1 - q_h \frac{du_j}{dv_j} &\leq 1 - q_h \epsilon < 1, \\ 1 - q_h \frac{du_j}{dv_j} &\geq 1 - q_h \lambda \geq 1 - \frac{1}{\lambda}\lambda = 0. \end{aligned}$$

Then, the quantities $1 - q_h du_j/dv_j$ and q_h always take a non-negative sign. Subsequently, considering the fact $\mathcal{S}(\dot{v}_j(t)) = \mathcal{S}(\dot{u}_{i,r}(t))$, (8.53) will induce that $\mathcal{S}(\dot{v}_{j+1}) = \mathcal{S}(\dot{u}_{i,r}(t))$.

On the other hand, it follows that $\mathcal{S}(\dot{v}_{j+1}(t)) = \mathcal{S}(\dot{u}_{j+1}(t))$ and $\mathcal{S}(\dot{v}_{i,r}(t)) = \mathcal{S}(\dot{u}_{i,r}(t))$ by the input-output monotonicity property. Hence, $\mathcal{S}(\dot{v}_{j+1}(t)) = \mathcal{S}(\dot{u}_{j+1}(t)) = \mathcal{S}(\dot{v}_{i,r}(t)) = \mathcal{S}(\dot{u}_{i,r}(t))$.

A.11: Proof of Theorem 5.2

Under each of the conditions (C_1), (C_2), (C_3'), and (C_4'), the desired output $u_{i,r}$ and its corresponding desired input $v_{i,r}$ take a same monotonicity. Thus, the concerned time interval $[0, T]$ can be divided as follows.

$$[0, T] = \{t_0\} \cup (t_0, t_1] \cup (t_1, t_2] \cup \cdots \cup (t_{n-1}, t_n], \tag{8.54}$$

where $t_0 = 0, t_n = T$, and $t_s, s = 1, \cdots, n-1$ are the extreme points of $u_{i,r}$ or $v_{i,r}$ satisfying $\dot{u}_{i,r}(t_s) = \dot{v}_{i,r}(t_s) = 0$. The principle idea of the convergence proof is briefly outlined below:

(1) first prove the learning convergence at the initial point $t = t_0 = 0$;

(2) assume that the learning convergence has been guaranteed from branch 1 to branch $k-1$. Prove the learning convergence for branch k. Then by induction, the learning convergence for every $1 \leq k \leq n$ can be derived.

Step 1: ILC convergence at the initial point $t = t_0$. Considering the i.i.c. for $z(t)$, namely $z(0) = 0$, the hysteresis input-output mapping at the initial point becomes

$$u(0) = \alpha k v(0). \tag{8.55}$$

Thus, in two consecutive two iterations

$$\begin{aligned} |\Delta u_j(0)| &= |u_{i,r}(0) - u_j(0)| \\ &= |(u_{i,r}(0) - u_{j-1}(0)) - (u_j(0) - u_{j-1}(0))| \\ &= |(u_{i,r}(0) - u_{j-1}(0)) - \alpha k(v_j(0) - v_{j-1}(0))| \\ &= |\Delta u_{j-1}(0) - \alpha k q_h \Delta u_{j-1}(0)| \\ &\leq |1 - \alpha k q_h||\Delta u_{j-1}(0)|, \end{aligned} \tag{8.56}$$

where the ILC updating law (5.29) is used. Noticing the gain restriction in (5.29), we have that $|1-\alpha k q_h| < 1$. Therefore, if letting $\rho = |1-\alpha k q_h|$ and $\sigma_j = 0$, the inequality (5.30) holds obviously.

Step 2: ILC convergence over the time interval $(t_{k-1}, t_k]$. Investigating from (5.24),

$$u(t) = u(v(t), v(t_{k-1}), z(t_{k-1}), \mathcal{S}(\dot{v}(t))), \quad t \in (t_{k-1}, t_k]. \tag{8.57}$$

Based on the learning convergence in branch $k-1$, the effect of initial condition error to the ILC convergence in the current branch can be ignored since what we concern is the asymptotical behavior of hysteresis after a sufficient large amount of iterations. Thus, we can write the hysteresis output as $u(v(t), \mathcal{S}(\dot{v}(t)))$ if no confusion occurs. This means that there are only two factors affecting the ultimate convergence of hysteresis: one is input $v(t)$ and the other one is its monotonicity.

Now, we are in the position of investigating the relationship of output errors in any two consecutive iterations. First assume that $\mathcal{S}(\dot{v}_{j-1}) \neq 0$ and the current iteration is in the j-th iteration. By the Mean Value Theorem

$$\begin{aligned}
& u_j - u_{j-1} \\
= \;& u(v_j, \mathcal{S}(\dot{v}_j)) - u(v_{j-1}, \mathcal{S}(\dot{v}_{j-1})) \\
= \;& u(v_j, \mathcal{S}(\dot{v}_j)) - u(v_j, \mathcal{S}(\dot{v}_{j-1})) \\
& + u(v_j, \mathcal{S}(\dot{v}_{j-1})) - u(v_{j-1}, \mathcal{S}(\dot{v}_{j-1})) \\
= \;& \sigma_j(t) + \frac{\partial u}{\partial v}(\bar{v}_{j-1}, \mathcal{S}(\dot{v}_{j-1}))(v_j - v_{j-1}),
\end{aligned} \tag{8.58}$$

where $\bar{v}_{j-1}$ lies in an interval determined by v_j and v_{j-1}, and

$$\sigma_j(t) = u(v_j, \mathcal{S}(\dot{v}_j)) - u(v_j, \mathcal{S}(\dot{v}_{j-1})).$$

Subsequently,

$$\begin{aligned}
|\Delta u_j| &= |u_{i,r} - u_j| \\
&= |(u_{i,r} - u_{j-1}) - (u_j - u_{j-1})| \\
&= |(u_{i,r} - u_{j-1}) \\
&\quad - \frac{\partial u}{\partial v}(\bar{v}_{j-1}, \mathcal{S}(\dot{v}_{j-1}))(v_j - v_{j-1}) - \sigma_j(t)\Big| \\
&\leq \left|1 - q_h \frac{\partial u}{\partial v}(\bar{v}_{j-1}, \mathcal{S}(\dot{v}_{j-1}))\right| |u_{i,r} - u_{j-1}| + |\sigma_j(t)| \\
&= \rho_{j-1}(t)|\Delta u_{j-1}| + |\sigma_j(t)|. \qquad (8.59)
\end{aligned}$$

where $\rho_{j-1}(t) = \left|1 - q_h \frac{\partial u}{\partial v}(\bar{v}_{j-1}, \mathcal{S}(\dot{v}_{j-1}))\right|$.

Considering the other possibility, if $\mathcal{S}(\dot{v}_{j-1}) = 0$ or equivalently $\dot{v}_{j-1} = 0$,

$$\begin{aligned}
\dot{u}_{j-1} &= \alpha k \dot{v}_{j-1} + (1-\alpha) D k \dot{z}_{j-1} \\
&= \alpha k \dot{v}_{j-1} + (1-\alpha) D k \dot{v}_{j-1} \times \\
&\qquad (D^{-1}(A - |z_{j-1}|^n(\gamma + \beta \mathcal{S}(\dot{v}_{j-1} z_{j-1})))) \\
&= 0,
\end{aligned}$$

and

$$\dot{v}_j = \dot{}_j x_1 + q_h(\dot{u}_{i,r} - \dot{u}_{j-1}) = q_h \dot{u}_{i,r} \neq 0.$$

Similar to (8.58) and (8.59), we have

$$u_j - u_{j-1} = \frac{\partial u}{\partial v}(\bar{v}_{j-1}, \mathcal{S}(\dot{v}_j))(v_j - v_{j-1}) + \sigma_j(t), \qquad (8.60)$$

and

$$|u_{i,r} - u_j| \leq \rho_{j-1}(t)|u_{i,r} - u_{j-1}| + |\sigma_j(t)|, \qquad (8.61)$$

where $\sigma_j(t) = u(v_{j-1}, \mathcal{S}(\dot{v}_j)) - u(v_{j-1}, \mathcal{S}(\dot{v}_{j-1}))$ and $\rho_{j-1}(t) = \left|1 - q_h \frac{\partial u}{\partial v}(\bar{v}_{j-1}, \mathcal{S}(\dot{v}_j))\right|$.

Applying *Lemma* 5.2, it can be seen that for any non-vanished sign function $\mathcal{S}(\cdot)$,

$$0 < \epsilon \leq \frac{\partial u}{\partial v}(\bar{v}_{j-1}, \mathcal{S}(\cdot)) \leq \lambda, \tag{8.62}$$

inducing that

$$\begin{aligned} \rho_{j-1}(t) &= 1 - q_h \frac{\partial u}{\partial v}(\bar{v}_{j-1}, \mathcal{S}(\cdot)) \\ &\leq 1 - q_h \epsilon \triangleq \rho < 1. \end{aligned} \tag{8.63}$$

The remaining work in this step is to prove $\sigma_j(t) \to 0$ as $j \to \infty$, which quantifies the effect of wrong estimation of hysteresis monotonicity to the learning convergence.

Without loss of generality, assume $\dot{v}_l \neq 0, l = 0, \cdots, j-1$. Otherwise, there exists $0 \leq l_0 \leq j-1$ such that $\dot{v}_{l_0} = 0$, and then $\dot{v}_{l_0+1} = q_h \dot{u}_{i,r} \neq 0$. By *Lemma* 5.3,

$$\mathcal{S}(\dot{v}_l) = \mathcal{S}(\dot{v}_{l+1}), \quad l = l_0 + 1, l_0 + 2, \cdots. \tag{8.64}$$

Hence, there exist at most l_0 pairs of $(\dot{v}_{l-1}, \dot{v}_l)$, $l \leq l_0 < \infty$, such that $\mathcal{S}(\dot{v}_l) \neq \mathcal{S}(\dot{v}_{l-1})$ and

$$\sigma_l(t) = u(\nu, \mathcal{S}(\dot{v}_l)) - u(\nu, \mathcal{S}(\dot{v}_{l-1})) \neq 0,$$

where $\nu = v_{l-1}$ or v_l.

Since $\dot{v}_l \neq 0, l = 0, \cdots, j-1$, differentiating the ILC law (5.29) will yield

$$\dot{v}_{l+1} = \delta_l \dot{v}_l + q_h \dot{u}_{i,r}. \tag{8.65}$$

where $\delta_l \triangleq 1 - q_h \frac{\dot{u}_l}{\dot{v}_l}$. Note that $0 \leq \delta_l < 1$ by *Lemma* 5.2. Iteratively, the relationship (8.65) gives that

$$\begin{aligned} \dot{v}_j &= \delta_{j-1}(\delta_{j-2}\dot{v}_{j-2} + q_h \dot{u}_{i,r}) + q_h \dot{u}_{i,r} \\ &\vdots \\ &= \Lambda_{1,j-1}\dot{v}_0 + \Lambda_{2,j-1} q_h \dot{u}_{i,r}, \end{aligned} \tag{8.66}$$

where

$$\Lambda_{1,j-1}=\prod_{l=0}^{j-1}\delta_l,\quad \Lambda_{2,j-1}=\sum_{l=0}^{j-1}\prod_{p=1}^{l}\delta_{j-p},$$

satisfying $0\leq\Lambda_{1,j-1}<1$ and $\Lambda_{2,j-1}\geq 1$. If $\mathcal{S}(\dot{v}_0)=\mathcal{S}(q_h\dot{u}_{i,r})$, then $\mathcal{S}\left(\dot{v}_j\right)=\mathcal{S}\left(\dot{v}_{j-1}\right)=\mathcal{S}(q_h\dot{u}_{i,r})$ for all j onward. If $\mathcal{S}(\dot{v}_0)\neq\mathcal{S}(q_h\dot{u}_{i,r})$, noticing that

$$\dot{v}_j=\Lambda_{2,j-1}\left(q_h\dot{u}_{i,r}+\frac{\Lambda_{1,j-1}}{\Lambda_{2,j-1}}\dot{v}_0\right),$$

$\mathcal{S}\left(\dot{v}_j\right)=\mathcal{S}(\dot{v}_0)$ as $\Lambda_{1,j-1}/\Lambda_{2,j-1}>-q_h\dot{u}_{i,r}/\dot{v}_0$ and $\mathcal{S}\left(\dot{v}_j\right)=\mathcal{S}(q_h\dot{u}_{i,r})$ as $\Lambda_{1,j-1}/\Lambda_{2,j-1}<-q_h\dot{u}_{i,r}/\dot{v}_0$. Since it is easy to see that the sequence $\{\Lambda_{1,j-1}\}_{j\in\mathbf{N}}$ is monotonically decreasing while the sequence $\{\Lambda_{2,j-1}\}_{j\in\mathbf{N}}$ is monotonically increasing, $\Lambda_{1,j-1}/\Lambda_{2,j-1}$ should be monotonically decreasing. Thus, there exists at most one pair of $(\dot{v}_{j_0-1},\dot{v}_{j_0})$, $j_0<\infty$ such that

$$\mathcal{S}\left(\dot{v}_{j_0}\right)\neq\mathcal{S}\left(\dot{v}_{j_0-1}\right),$$

namely, v_{j_0} and v_{j_0-1} take different monotonicities at the same time instant. In the sequel, there exists one and only one $j_0<\infty$ such that $\sigma_{j_0}(t)\neq 0$.

Summarily, we can conclude that the function $\sigma_j(t)$ will vanish after finite iterations, which obviously means that $\sigma_j\to 0$ as $j\to\infty$, which complete the learning convergence proof in the k-th branch.

A.12: Proof of Theorem 6.1

The following time weighted CEF is employed.

$$\begin{aligned}E_i(t) &= \frac{1}{2}e^{-\zeta t}e_i^2(t)+\frac{1}{2q}\int_0^t e^{-\zeta\tau}\left(\Delta u_{i,r}\right)^2 d\tau\\ &\quad+\frac{1}{2q}\int_0^t e^{-\zeta\tau}\left(\Delta u_i\right)^2 d\tau,\end{aligned}\tag{8.67}$$

where

$$\zeta>2L_\eta+2q+\frac{8q\rho^2}{1-\rho^2}\tag{8.68}$$

is a finite positive constant, ρ is from (5.30), and $u_{i,r}$ is from (5.31). Here, $e_i = x_r - x_i$, $\Delta u_i = u_{i,r} - u_i$, and $\Delta u_{i,r} = u_r - u_{i,r}$, where u_r is the desired hysteresis output that leads to the desired state output x_r and $u_{i,r}$ is generated by ILC law (5.31). First, differencing $E_i(t)$ yields

$$\begin{aligned}
\Delta E_i &= E_i - E_{i-1} \\
&= \frac{1}{2}e^{-\zeta t}e_i^2(t) - \frac{1}{2}e^{-\zeta t}e_{i-1}^2(t) \\
&\quad + \frac{1}{2q}\int_0^t e^{-\zeta\tau}\left[(\Delta u_{i,r})^2 - (\Delta u_{i-1,r})^2\right]d\tau \\
&\quad + \frac{1}{2q}\int_0^t e^{-\zeta\tau}\left[(\Delta u_i)^2 - (\Delta u_{i-1})^2\right]d\tau. \qquad (8.69)
\end{aligned}$$

Each term on the right hand side of (8.69) is bounded separately. Using the error dynamics (5.11), the first term on the right hand side of (8.69) can be bounded as follows

$$\begin{aligned}
&\frac{1}{2}e^{-\zeta t}e_i^2(t) \\
\leq\ & -\frac{\zeta - 2L_\eta}{2}\int_0^t e^{-\zeta\tau}e_i^2(\tau)d\tau \\
&+ \int_0^t e^{-\zeta\tau}e_i(\tau)\left[(u_r - u_{i,r}) + (u_{i,r} - u_{i-1,r})\right]d\tau \\
&+ \int_0^t e^{-\zeta\tau}e_i(\tau)(u_{i-1,r} - u_i)d\tau \\
\leq\ & -\frac{\zeta - 2L_\eta - 2q}{2}\int_0^t e^{-\zeta\tau}e_i^2(\tau)d\tau \\
&+ \int_0^t e^{-\zeta\tau}e_i(\tau)\left(u_r - u_{i,r}\right)d\tau \\
&+ \int_0^t e^{-\zeta\tau}e_i(\tau)\left(u_{i-1,r} - u_i\right)d\tau. \qquad (8.70)
\end{aligned}$$

For the third term on the right hand side of (8.69), using the algebraic relationship $(a-b)^2 - (a-c)^2 = -2(a-b)(b-c) - (b-c)^2$ where a, b, c are scalars and the ILC law (5.31),

$$\frac{1}{2q}\int_0^t e^{-\zeta\tau}\left[(\Delta u_{i,r})^2 - (\Delta u_{i-1,r})^2\right]d\tau$$

$$
\begin{aligned}
&= \frac{1}{2}\int_0^t e^{-\zeta\tau} e_i(\tau)\left[2u_{i,r} - 2u_r\right] d\tau - \frac{q}{2}\int_0^t e^{-\zeta\tau} e_i^2(\tau) d\tau \\
&= \int_0^t e^{-\zeta\tau} e_i(\tau)\left[u_{i,r} - u_r\right] d\tau - \frac{q}{2}\int_0^t e^{-\zeta\tau} e_i^2(\tau) d\tau. \qquad (8.71)
\end{aligned}
$$

Now consider the last term on the right hand side of (8.69). From the inequality (5.30) in *Theorem* 5.2, it follows that

$$
\rho^2 |\Delta u_{i-1}|^2 \geq |\Delta u_i|^2 - 2\rho |\Delta u_{i-1}| \cdot |\sigma_i| - |\sigma_i|^2. \qquad (8.72)
$$

In the sequel, it yields that

$$
\begin{aligned}
& (\Delta u_i)^2 - (\Delta u_{i-1})^2 \\
= \; & (u_{i,r} - u_i)^2 - (u_{i-1,r} - u_i)^2 \\
& + (u_{i-1,r} - u_i)^2 \\
& - (u_{i-1,r} - u_{i-1})^2 \\
\leq \; & (u_{i,r} - u_{i-1,r})(u_{i,r} + u_{i-1,r} - 2u_i) \\
& - \frac{1-\rho^2}{\rho^2}(u_{i-1,r} - u_i)^2 \\
& + \frac{2}{\rho}|u_{i-1,r} - u(v_i, \mathcal{S}(\dot{v}_i), t)| \cdot |\sigma_i| + \frac{1}{\rho^2}|\sigma_i|^2 \\
= \; & qe_i \cdot (2u_{i-1,r} + qe_i - 2u_i) \\
& - \frac{1-\rho^2}{2\rho^2}(u_{i-1,r} - u_i)^2 \\
& - \frac{1-\rho^2}{2\rho^2}\Big[(u_{i-1,r} - u_i)^2 \\
& - \frac{4\rho}{1-\rho^2}|u_{i-1,r} - u_i| \cdot \\
& |\sigma_i| + \left(\frac{2\rho}{1-\rho^2}|\sigma_i|\right)^2\Big] + \frac{1+\rho^2}{(1-\rho^2)\rho^2}|\sigma_i|^2 \\
\leq \; & qe_i \cdot (2u_{i-1,r} + qe_i - 2u_i) \\
& - \frac{1-\rho^2}{2\rho^2}(u_{i-1,r} - u_i)^2 \\
& + \frac{1+\rho^2}{(1-\rho^2)\rho^2}|\sigma_i|^2 \qquad (8.73)
\end{aligned}
$$

Consequently,

$$\begin{aligned}
& \frac{1}{2q}\int_0^t e^{-\zeta\tau}\left[(\Delta u_i)^2-(\Delta u_{i-1})^2\right]d\tau \\
\leq\ & \int_0^t e^{-\zeta\tau}e_i(\tau)\left(u_{i-1,r}-u_i\right)d\tau \\
& +\frac{q}{2}\int_0^t e^{-\zeta\tau}e_i^2(\tau)d\tau \\
& -\frac{1}{2q}\int_0^t e^{-\zeta\tau}\frac{1-\rho^2}{2\rho^2}\left(u_{i-1,r}-u_i\right)^2 d\tau \\
& +\frac{1}{2q}\int_0^t e^{-\zeta\tau}\frac{1+\rho^2}{(1-\rho^2)\,\rho^2}|\sigma_i|^2 d\tau. \qquad (8.74)
\end{aligned}$$

Substituting (8.70), (8.71) and (8.74) into (8.69) yields

$$\begin{aligned}
\Delta E_i \leq\ & -\frac{\zeta-2L_\eta-2q}{2}\int_0^t e^{-\zeta\tau}(e_i^2(\tau))d\tau \\
& +2\int_0^t e^{-\zeta\tau}e_i(\tau)(u_{i-1,r}-u_i d\tau \\
& -\frac{1}{2q}\int_0^t e^{-\zeta\tau}\frac{1-\rho^2}{2\rho^2}\left(u_{i-1,r}-u_i\right)^2 d\tau \\
& -\frac{1}{2}e^{-\zeta t}(e_{i-1}^2(t)) \\
& +\frac{1}{2q}\int_0^t e^{-\zeta\tau}\frac{1+\rho^2}{(1-\rho^2)\,\rho^2}|\sigma_i|^2 d\tau. \qquad (8.75)
\end{aligned}$$

Note that the inequality (8.68) and the following relationship

$$\begin{aligned}
& 2e_i(\tau)\left(u_{i-1,r}-u_i\right) \\
\leq\ & \frac{4q\rho^2}{1-\rho^2}e_i^2(\tau)+\frac{1-\rho^2}{\rho^2}\frac{1}{4q}\left(u_{i-1,r}-u_i\right)^2,
\end{aligned}$$

we have

$$\Delta E_i \leq -\frac{1}{2}e^{-\zeta t}e_{i-1}^2(t)+\frac{1}{2q}\int_0^t e^{-\zeta\tau}\frac{1+\rho^2}{(1-\rho^2)\,\rho^2}|\sigma_i|^2 d\tau,$$

which implies that

$$\begin{aligned}
E_i &= E_0+\sum_{p=1}^{i}\Delta E_p \\
&\leq E_0-\frac{1}{2}e^{-\zeta t}\sum_{p=0}^{i-1}e_p^2(t)
\end{aligned}$$

$$+\left(\frac{1+\rho^2}{2q\left(1-\rho^2\right)\rho^2}\right)\int_0^t e^{-\zeta\tau}\sum_{p=1}^{i}|\sigma_p(\tau)|^2 d\tau. \tag{8.76}$$

Note that E_i is positive, E_0 is finite because e_0, $\Delta u_{0,r}$, and Δu_0 are finite, and $\lim_{p\to\infty}\sigma_p = 0$. For any small $\iota > 0$, there must exist a finite iteration number i_ι such that the output tracking error $|e_i(t)| < \iota$ for all $i \geq i_\iota$ and $t \in [0, T]$. This completes the proof.

A.13: Proof of Theorem 5.4

Differentiating the learning law (5.36), it yields that

$$\begin{aligned}
& \dot{v}_j(t) \\
= \;& (1-\zeta_0)\dot{v}_{j-1}(t) + q_h(\dot{u}_r(t) - \dot{u}_{j-1}(t)) \\
= \;& (1-\zeta_0)\dot{v}_{j-1}(t) + q_h\dot{u}_r(t) - q_h \\
& \times\left(kA - \frac{|u_{j-1}|^n}{k^{n-1}D^n}(\gamma + \beta\mathcal{S}(\dot{v}_{j-1}u_{j-1}))\right)\dot{v}_{j-1}(t).
\end{aligned}$$

Subsequently,

$$\dot{v}_j(t) = \Theta_{j-1}\dot{v}_{j-1}(t) + q_h\dot{u}_r(t),$$

where

$$\Theta_{j-1} \triangleq 1 - \zeta_0 - q_h\left(kA - \frac{|u_{j-1}|^n}{k^{n-1}D^n}(\gamma + \beta\mathcal{S}(\dot{v}_{j-1}u_{j-1}))\right).$$

Noticing that $0 \leq q_h(kA - \frac{|u|^n}{k^{n-1}D^n}(\gamma + \beta\mathcal{S}(\dot{v}u))) \leq 1 - \zeta_0$, we have

$$0 \leq \Theta_{j-1} \leq 1 - \zeta_0.$$

Due to the relationship $\mathcal{S}(\dot{v}_r) = \mathcal{S}(q_h\dot{u}_r)$ as $\dot{u}_r \neq 0$, the correct monotonicity for input v can be learned within finite iterations, as discussed in (8.66). As $\dot{u}_{i,r} = 0$,

$$\dot{v}_j(t) = \Theta_{j-1}\dot{v}_{j-1}(t).$$

Although the correct monotonicity of input v may not be learned in finite iterations, the relationship $\mathcal{S}(\dot{v}_j) = \mathcal{S}(\dot{v}_{j-1})$ always holds as $j < \infty$, due to $\Theta_{j-1} > 0$. In either of the two cases, therefore, $\mathcal{S}(\dot{v}_j) = \mathcal{S}(\dot{v}_{j-1})$ after certain finite iterations. Then, we can ignore the effect of input monotonicity to learning convergence and write the solution of Eq. (5.35) as

$$u(t) = u(v(t), v(t_s), u(t_s)), \quad t \in [t_s, t_{s+1}] \subset [0, T],$$

in each monotone branch of $v(t), t \in [t_s, t_{s+1}]$, when considering the asymptotical convergence property of the system only.

To achieve the output convergence in such singular case, the main idea here is still similar as in the normal cases: consider the learning convergence in each monotone branch of hysteresis separately; the analysis in current branch is based on the convergence result in the previous adjacent branch.

Step 1: Learning convergence in the first monotone branch. First prove that the operator, induced by the ILC law (5.36),

$$\mathcal{T}[v(t)] = (1 - \zeta_0)v(t) + q_h \Delta u(t), \tag{8.77}$$

is a contraction operator in the space $C^1([t_0, t_1], \mathbf{R}, \|\cdot\|)$, where $t_0 = 0$.

When $u_r(t), v(t) \in C^1([t_0, t_1], \mathbf{R}, \|\cdot\|)$, according to *Lemma* 5.1, $z(t) \in C^1([t_0, t_1], \mathbf{R}, \|\cdot\|)$. In the sequel, $u(t) \in C^1([t_0, t_1], \mathbf{R}, \|\cdot\|)$ and then $\Delta u(t) \in C^1([t_0, t_1], \mathbf{R}, \|\cdot\|)$. From (8.77), $\mathcal{T}$ is an operator which maps the elements of the Banach space $C^1([t_0, t_1], \mathbf{R}, \|\cdot\|)$ into itself.

Considering the i.i.c. for v and u separately, for any $v_s, \in C^1([t_0, t_1], \mathbf{R}, \|\cdot\|), s = 1, 2$, the corresponding output is

$$u_s(t) = u(v_s(t), v_s(t_0), u_s(t_0)) = u(v_s(t), \xi_v, u_r(0)).$$

Thus,

$$\begin{aligned}
& |\mathcal{T}[v_1(t)] - \mathcal{T}[v_2(t)]| \\
= \; & |(1-\zeta_0)(v_1(t)-v_2(t)) - q_h(u_1(t)-u_2(t))| \\
= \; & |(1-\zeta_0)(v_1(t)-v_2(t)) - q_h(u(v_1(t),\xi_v,u_r(0)) - u(v_2(t),\xi_v,u_r(0)))| \\
= \; & \left|(1-\zeta_0)(v_1(t)-v_2(t)) - q_h\frac{\partial u}{\partial v}(\bar{v})(v_1(t)-v_2(t))\right| \\
\leq \; & \left|(1-\zeta_0) - q_h\frac{\partial u}{\partial v}(\bar{v})\right| |v_1(t)-v_2(t)| \qquad (8.78)
\end{aligned}$$

where $\bar{v}(t)$ is lied in the interval (v_1, v_2) or (v_2, v_1) by the Mean Value Theorem. Noticing that $q_h \partial u/\partial v \geq 0$ and $|\partial u/\partial v| \leq \lambda$, and considering the gain restriction (5.38), it is easy to see that

$$\left|(1-\zeta_0) - q_h\frac{\partial u}{\partial v}(\bar{v})\right| < 1,$$

that is, $\mathcal{T}$ is indeed a contaction operator in the Banach space $C^1([t_0,t_1],\mathbf{R},\|\cdot\|)$. According to the Banach fixed-point theorem, $\mathcal{T}$ has a unique fixed point $v_1^*(t) \in C^1([t_0,t_1],\mathbf{R},\|\cdot\|)$, and the input sequence, determined by (5.36), will converge to this point.

Since $v_1^* = \mathcal{T}[v_1^*]$, substituting $v = v_1^*$ into (8.77), we finally have

$$\lim_{j\to\infty} |\Delta u_j(t)| = \frac{\zeta_0}{|q_h|}|v_1^*(t)| \leq \frac{\zeta_0}{|q_h|}|v_1^*(t)|_s \qquad (8.79)$$

where $|v_1^*(t)|_s$ denotes the supreme norm of $v_1^*(t)$ as $t \in [t_0, t_1]$.

Step 2: Learning convergence in the k-th monotone branch. Assuming the existence of the fixed-point input function $v_{k-1}^*(t), t \in [t_{k-2}, t_{k-1}]$ for the $(k-1)$-th branch, the output function

$$u_{k-1}^*(t) = u(v_{k-1}^*(t), v_{k-2}^*(t_{k-2}), u_r(t_{k-2})), \quad t \in [t_{k-2}, t_{k-1}]$$

is also fixed. By the same reason we presented in the normal cases, the output solution in the current hysteresis branch can be simply written as

$$u(t) = u(v(t), v^*_{k-1}(t_{k-1}), u^*_{k-1}(t_{k-1})), \quad t \in [t_{k-1}, t_k], \tag{8.80}$$

namely, the initial condition effect to the ILC convergence in $[t_{k-1}, t_k]$ is ignored when the asymptotical behavior of hysteresis along the iteration axis is concerned only. Similar to the discussion in (8.78), (8.77) also defines a contraction operator $\mathcal{T}$ in $[t_{k-1}, t_k]$, and its fixed point is $v^*_k(t) \in C^1([t_{k-1}, t_k], \mathbf{R}, \|\cdot\|)$. The input sequence, determined by (5.36), will converge to v^*_k and the following relationship holds

$$\lim_{j\to\infty} |\Delta u_j(t)| = \frac{\zeta_0}{|q_h|}|v^*_k(t)| \le \frac{\zeta_0}{|q_h|}|v^*_k(t)|_s \tag{8.81}$$

as $t \in [t_{k-1}, t_k]$.

Step 3: Learning convergence over $[0, T]$. Define a new function $v^*(t)$ as follows:

$$v^*(t) = v^*_k(t), \quad \text{if } \ t \in [t_{k-1}, t_k], \quad k = 1, \cdots, n. \tag{8.82}$$

Obviously, (8.79) and (8.81) give that

$$\lim_{j\to\infty} |\Delta u_j(t)| = \frac{\zeta_0}{|q_h|}|v^*(t)| \le \frac{\zeta_0}{|q_h|}|v^*(t)|_s, \ t \in [0, T]. \tag{8.83}$$

It is worthy of noticing that $|\Delta u_j| = \rho|\Delta u_{j-1}| + (|\Delta u_j| - \rho|\Delta u_{j-1}|)$, where $0 < \rho < 1$. Let $\sigma_j(t) = |\Delta u_j| - \rho|\Delta u_{j-1}|$, and then (8.83) implies that

$$\lim_{j\to\infty} \sigma_j(t) = (1-\rho) \lim_{j\to\infty} |\Delta u_j(t)| \le \frac{\zeta_0(1-\rho)}{|q_h|}|v^*(t)|_s. \tag{8.84}$$

This completes the proof.

A.14: Proof of Theorem 5.5

Similar to the first singular case, as $t \in \Omega_1$

$$\dot{v}_j(t) = \Theta_{j-1}\dot{v}_{j-1}(t) + q^0_h \dot{u}_r(t),$$

where

$$\Theta_{j-1} \triangleq 1 - \zeta_0 - q_h^0 \frac{\dot{u}_{j-1}}{\dot{v}_{j-1}}.$$

Noticing that $-\zeta_0/2 \leq q_h^0 \frac{\dot{u}_{j-1}}{\dot{v}_{j-1}} \leq \zeta_0/2$, we have

$$0 < 1 - \frac{3\zeta_0}{2} \leq \Theta_{j-1} \leq 1 - \frac{\zeta_0}{2} < 1.$$

Then, the relationship $\mathcal{S}(\dot{v}_j) = \mathcal{S}(\dot{v}_{j-1})$ holds after certain finite iterations. Subsequently, the operator, induced by the ILC law (5.44),

$$\mathcal{T}[v(t)] = (1 - \zeta_0)v(t) + q_h^0 \Delta u(t), \tag{8.85}$$

is a contraction operator in the space $C^1([0,T], \mathbf{R}, \|\cdot\|)$, and there exists a unique fixed input function v^* such that $v^* = \mathcal{T}[v^*]$. Thus, a bound for the output tracking error can be easily derived as in the preceding part

$$\lim_{j\to\infty} |\Delta u_j(t)| \leq \frac{\zeta_0}{|q_h^0|} |v^*(t)|_s, \quad t \in \Omega_1. \tag{8.86}$$

Next analyze the boundedness of the output tracking error in $\Omega_2 = [0,T] - \Omega_1$, where Ω_2 is composed of a number of open sets, each covering a singular point t_s with its length δ. In each interval $(t_s - \delta/2, t_s + \delta/2)$ of Ω_2, denote u^* the system state corresponding to v^*. Then,

$$\begin{aligned} |u_r(t) - u^*(t)| &\leq |u_r(t) - u_r(t_s - \delta/2)| \\ &\quad + |u_r(t_s - \delta/2) - u^*(t_s - \delta/2)| \\ &\quad + |u^*(t) - u^*(t_s - \delta/2)|. \end{aligned}$$

Considering the C^1 boundedness of u_r and applying the Mean Value Theorem,

$$\begin{aligned} |u_r(t) - u_r(t_s - \delta/2)| &\leq |\dot{u}_r(\bar{t})||t - t_s + \delta/2| \\ &\leq \beta_1 \delta \end{aligned} \tag{8.87}$$

where $|\dot{u}_r(t)| \leq \beta_1, t \in \Omega_2$ for certain finite constant β_1. On the other hand, the C^1 property of v^* in $[0, T]$ also implies the C^1 boundedness of u^* by *Lemma* 5.1. Subsequently, there exists another constant β_2 such that $|\dot{u}^*(t)| \leq \beta_2, t \in \Omega_2$ and

$$|u^*(t) - u^*(t_s - \delta/2)| \leq \beta_2 |t - t_s + \delta/2| \leq \beta_2 \delta. \tag{8.88}$$

Moreover, note that $|u_r(t_s - \delta/2) - u^*(t_s - \delta/2)| \leq \frac{\zeta_0}{|q_h^0|}|v^*(t)|_s$ by (8.86). Finally, we have that

$$\lim_{j \to \infty} |\Delta u_j(t)| \leq \frac{\zeta_0}{|q_h^0|}|v^*(t)|_s + \delta \sum_{i=1}^{2} \beta_i, \quad t \in \Omega_2. \tag{8.89}$$

Let $\sigma_j(t) = |\Delta u_j| - \rho|\Delta u_{j-1}|$, satisfying $|\Delta u_j| = \rho|\Delta u_{j-1}| + \sigma_j(t)$, and then (8.87) and (8.89) imply (5.46) directly.

A.15: Proof of Property 6.1

Integrating $F_1(\bar{\mathbf{c}}(z), \bar{v}(z)) = \mathbf{0}$, i.e. $F_1(\bar{\mathbf{c}}(z), \bar{u}) = \mathbf{0}$, along the spatial coordinate from 0 to z,

$$\begin{aligned} &-B\bar{u}\bar{\mathbf{c}}(z) + D\frac{\partial \bar{\mathbf{c}}(z)}{\partial z} + \left(B\bar{u}\bar{\mathbf{c}}(0) - D\frac{\partial \bar{\mathbf{c}}(0)}{\partial z}\right) \\ &+ \int_0^z \mathbf{f}(\bar{\mathbf{c}}(\tau), \tau) d\tau = \mathbf{0}. \end{aligned} \tag{8.90}$$

Further integrating (8.90),we have that

$$\begin{aligned} &-B\bar{u}\int_0^z \bar{\mathbf{c}}(\tau) d\tau + D(\bar{\mathbf{c}}(z) - \bar{\mathbf{c}}(0)) \\ &+ \left(B\bar{u}\bar{\mathbf{c}}(0) - D\frac{\partial \bar{\mathbf{c}}(0)}{\partial z}\right) z \\ &+ \int_0^z \int_0^\tau \mathbf{f}(\bar{\mathbf{c}}(\zeta), \zeta) d\zeta d\tau = \mathbf{0}, \end{aligned} \tag{8.91}$$

or equivalently

$$-B\bar{u}\int_0^z \bar{\mathbf{c}}(\tau) d\tau + D(\bar{\mathbf{c}}(z) - \bar{\mathbf{c}}(0))$$

$$+\left(B\bar{u}\bar{\mathbf{c}}(0)-D\frac{\partial\bar{\mathbf{c}}(0)}{\partial z}\right)z$$
$$+\int_0^z(z-\tau)\mathbf{f}(\bar{\mathbf{c}}(\tau),\tau)d\tau=\mathbf{0}. \tag{8.92}$$

Write (8.92) with the following form,

$$\begin{aligned}\bar{\mathbf{c}}(z) &= \bar{\mathbf{c}}(0)+D^{-1}B\bar{u}\int_0^z\bar{\mathbf{c}}(\tau)d\tau\\ &\quad-\left(D^{-1}B\bar{u}\bar{\mathbf{c}}(0)-\frac{\partial\bar{\mathbf{c}}(0)}{\partial z}\right)z\\ &\quad-D^{-1}\int_0^z(z-\tau)\mathbf{f}(\bar{\mathbf{c}}(\tau),\tau)d\tau.\end{aligned} \tag{8.93}$$

Noticing the Lipschitz condition for $\mathbf{f}(\bar{\mathbf{c}}(z),z)$, the velocity restriction $\bar{u}\in[v_{min},v_{max}]$, and taking norm on both sides of (8.93), we can see that

$$\begin{aligned}\|\bar{\mathbf{c}}(z)\| &\le \left(1+v_{max}z\left\|D^{-1}B\right\|\right)\|\bar{\mathbf{c}}(0)\|\\ &\quad+z\left\|\frac{\partial\bar{\mathbf{c}}(0)}{\partial z}\right\|\\ &\quad+\|D^{-1}B\|v_{max}\int_0^z\|\bar{\mathbf{c}}(\tau)\|d\tau\\ &\quad+\|D^{-1}\|\int_0^z(z-\tau)\omega_{\mathbf{f}}(\tau)\|\bar{\mathbf{c}}(\tau)\|\,d\tau.\end{aligned} \tag{8.94}$$

Since $\|D^{-1}B\|v_{max}+\|D^{-1}\|(z-\tau)\omega_{\mathbf{f}}(\tau)\geq 0$ and $\left(1+v_{max}z\left\|D^{-1}B\right\|\right)\|\bar{\mathbf{c}}(0)\|+z\left\|\frac{\partial\bar{\mathbf{c}}(0)}{\partial z}\right\|$ is nondecreasing in z, applying the generalized Gronwall inequality [**?**] to (8.94) yields that

$$\begin{aligned}&\|\bar{\mathbf{c}}(z)\|\\ \le\ &\left(\left(1+v_{max}z\left\|D^{-1}B\right\|\right)\|\bar{\mathbf{c}}(0)\|+z\left\|\frac{\partial\bar{\mathbf{c}}(0)}{\partial z}\right\|\right)\\ &\times\exp\left(\int_0^z\left[v_{max}\|D^{-1}B\|\right.\right.\\ &\quad\left.\left.+\|D^{-1}\|(z-\tau)\omega_{\mathbf{f}}(\tau)\right]d\tau\right).\end{aligned} \tag{8.95}$$

A.16: Proof of Theorem 6.1

Letting $z = L$ in (8.93),

$$\begin{aligned}\bar{\mathbf{c}}(L) &= \bar{\mathbf{c}}(0) + \frac{\partial \bar{\mathbf{c}}(0)}{\partial z} L \\ &\quad + \bar{u} D^{-1} B \left(\int_0^L \bar{\mathbf{c}}(z) dz - \bar{\mathbf{c}}(0) L \right) \\ &\quad - D^{-1} \int_0^L (L - z) \mathbf{f}(\bar{\mathbf{c}}(z), z) dz. \end{aligned} \tag{8.96}$$

By Assumption 6.3, corresponding to the desired steady state output y^*, a unique pair of $(\bar{u}^*, \bar{\mathbf{c}}^*)$ exists, and satisfies

$$\begin{aligned}\bar{\mathbf{c}}^*(L) &= \bar{\mathbf{c}}^*(0) + \frac{\partial \bar{\mathbf{c}}^*(0)}{\partial z} L \\ &\quad + D^{-1} B \bar{u}^* \left(\int_0^L \bar{\mathbf{c}}^*(z) dz - \bar{\mathbf{c}}^*(0) L \right) \\ &\quad - D^{-1} \int_0^L (L - z) \mathbf{f}(\bar{\mathbf{c}}^*(z), z) dz. \end{aligned}$$

Let $\Delta \bar{\mathbf{c}}(z) = \bar{\mathbf{c}}^*(z) - \bar{\mathbf{c}}(z)$ and $\Delta \bar{u} = \bar{u}^* - \bar{u}$. Then, $\Delta \bar{\mathbf{c}}_i(0) = \mathbf{0}$ and $\partial \Delta \bar{\mathbf{c}}_i(0)/\partial z = \mathbf{0}$ in ith iteration due to the strict repeatable assumption of process. Subsequently, the relationship of input/output errors in ith iteration can be given by the following inequality:

$$\begin{aligned} & \|\Delta \bar{\mathbf{c}}_i(L)\| \\ = & \left\| D^{-1} B \bar{u}^* \left(\int_0^L \bar{\mathbf{c}}^*(z) dz - \bar{\mathbf{c}}^*(0) L \right) \right. \\ & \quad - D^{-1} \int_0^L (L - z) \mathbf{f}(\bar{\mathbf{c}}^*(z), z) dz \\ & \quad - D^{-1} B \bar{u}_i \left(\int_0^L \bar{\mathbf{c}}_i(z) dz - \bar{\mathbf{c}}_i(0) L \right) \\ & \quad \left. + D^{-1} \int_0^L (L - z) \mathbf{f}(\bar{\mathbf{c}}_i(z), z) dz \right\| \\ \leq & \; v_{max} \|D^{-1} B\| \int_0^L \|\Delta \bar{\mathbf{c}}_i(z)\| dz \\ & \quad + \|D^{-1} B\| \left\| \int_0^L \bar{\mathbf{c}}_i(z) dz - \bar{\mathbf{c}}_i(0) L \right\| |\Delta \bar{u}_i| \end{aligned} \tag{8.97}$$

$$+ \|D^{-1}\| \int_0^L (L-z) \times \|\mathbf{f}(\bar{\mathbf{c}}^*(z), z) - \mathbf{f}(\bar{\mathbf{c}}_i(z), z)\| dz.$$

Using the boundedness property of $\bar{\mathbf{c}}(z)$ and the Lipschitz condition of $\mathbf{f}(\bar{\mathbf{c}}(z), z)$, it is easy to see that

$$\begin{aligned} & \|\Delta\bar{\mathbf{c}}_i(L)\| \\ \leq\ & \|D^{-1}B\| \left(\int_0^L \|\Xi_0(z)\| dz + \|\bar{\mathbf{c}}^*(0)\| L \right) |\Delta\bar{u}_i| \\ & + \int_0^L \left(v_{max} \|D^{-1}B\| + \|D^{-1}\| (L-z) \omega_{\mathbf{f}}(z) \right) \\ & \times \|\Delta\bar{\mathbf{c}}_i(z)\| dz. \end{aligned} \tag{8.98}$$

Similar to the proof in Property 6.1, we get by using the generalized Gronwall inequality that

$$\|\Delta\bar{\mathbf{c}}_i(L)\| \leq \Xi_1 |\Delta\bar{u}_i|. \tag{8.99}$$

where Ξ_1 is given in (6.18).

Now, assume $\Delta\bar{y}_i = y^* - \bar{y}_i$. By the global Lipschitz condition of function h and (8.99), the input/output errors satisfy

$$|\Delta\bar{y}_i| \leq \omega_h \|\Delta\bar{\mathbf{c}}_i(L)\| \leq \lambda |\Delta\bar{u}_i| \tag{8.100}$$

with $\lambda = \omega_h \Xi_1$, where ω_h is the Lipschitz constant given in (6.11). The value of λ quantifies the input-output gradient, and the input-output inequality (8.100) is important for us to prove the convergence of IBLC.

Considering the steady-state input errors $\Delta\bar{u}_i$ in two consecutive iterations, we have that

$$|\Delta\bar{u}_{i+1}| \ = \ |\bar{u}^* - \bar{u}_{i+1}|$$

$$\begin{aligned} &= |(\bar{u}^* - \bar{u}_i) - (\bar{u}_{i+1} - \bar{u}_i)| \\ &= |\Delta\bar{u}_i - \rho\Delta\bar{y}_i|, \end{aligned} \tag{8.101}$$

where the IBLC law (6.16) is applied. Applying the Differential Mean Value Theorem to function $\bar{y}(\bar{u})$ gives that

$$\begin{aligned} \Delta\bar{y}_i &= y^* - \bar{y}_i = \bar{y}(\bar{u}^*) - \bar{y}(\bar{u}_i) \\ &= \frac{d\bar{y}(\zeta)}{d\bar{u}}(\bar{u}^* - \bar{u}_i) = \frac{d\bar{y}(\zeta)}{d\bar{u}}\Delta\bar{u}_i, \end{aligned} \tag{8.102}$$

where ζ lies in the interval $[\bar{u}^*, \bar{u}_i]$ or $[\bar{u}_i, \bar{u}^*]$. Notice that Assumption 6.3 implies the strict monotonicity of input-output relationship, that is, for all $\bar{u} \in [v_{min}, v_{max}]$

$$\frac{d\bar{y}(\bar{u})}{d\bar{u}} = \left(\frac{\partial h}{\partial \bar{\mathbf{c}}}\right)^T \frac{\partial \bar{\mathbf{c}}}{\partial \bar{u}}\bigg|_{z=L} > (<)0.$$

If we choose the learning gain ρ such that $\text{sign}(\rho) = \text{sign}\left(\frac{d\bar{y}(\bar{u})}{d\bar{u}}\right)$, then $\text{sign}\left(\rho\frac{d\bar{y}(\bar{u})}{d\bar{u}}\right) = 1$ for any $\bar{u} \in [v_{min}, v_{max}]$. Subsequently, multiplying by ρ and then taking sign operations on both sides of (8.102) yield that

$$\begin{aligned} \text{sign}\left(\rho\Delta\bar{y}_i\right) &= \text{sign}\left(\rho\frac{d\bar{y}(\zeta)}{d\bar{u}}\Delta\bar{u}_i\right) \\ &= \text{sign}\left(\rho\frac{d\bar{y}(\zeta)}{d\bar{u}}\right)\text{sign}\left(\Delta\bar{u}_i\right) \\ &= \text{sign}\left(\Delta\bar{u}_i\right). \end{aligned} \tag{8.103}$$

Thus, we can derive from (8.101) and (8.103) that

$$\begin{aligned} |\Delta\bar{u}_{i+1}| &= |\Delta\bar{u}_i - \rho\Delta\bar{y}_i|, \\ &= ||\Delta\bar{u}_i| - |\rho||\Delta\bar{y}_i|| \\ &\leq |1 - \lambda|\rho|| \cdot |\Delta\bar{u}_i|, \end{aligned} \tag{8.104}$$

where the inequality (8.100) is adopted. Noticing the gain range (6.17) for the learning gain ρ, it is easy to see that

$$|\Delta\bar{u}_{i+1}| \leq \delta|\Delta\bar{u}_i| < |\Delta\bar{u}_i|. \tag{8.105}$$

where $0 < \delta < 1$ is given in (6.17). Thus, $\Delta\bar{u}_i$ and then $\Delta\bar{y}_i$ will converge to zero as $i \to \infty$.

According to Assumption 6.4, $\|\mathbf{c}_i(L,t) - \bar{\mathbf{c}}_i(L)\| < \epsilon$ as $t \in [T_\epsilon, T]$ for each iteration i. In the sequel, $y_i(t)$ will enter into its steady state stage with an error less than $\omega_h\epsilon$ in each trial, i.e., $|y_i(t) - \bar{y}_i| < \omega_h\epsilon$ as $t \in [T_\epsilon, T]$. Thus,

$$\begin{aligned}\lim_{i\to\infty} |y_i(t) - y^*| &\leq \lim_{i\to\infty} (|\bar{y}_i - y^*| + |y_i(t) - \bar{y}_i|) \\ &= \lim_{i\to\infty} (|\Delta\bar{y}_i| + |y_i(t) - \bar{y}_i|) \\ &\leq \omega_h\epsilon\end{aligned}$$

as $t \in [T_\epsilon, T]$.

A.17: Proof of Theorem 6.2

As can be seen from (8.100), $|\Delta\bar{y}_i| \leq \lambda|\Delta\bar{u}_i|$, and then there exists a quantity $0 < \lambda_i \leq \lambda$ such that

$$|\Delta\bar{y}_i| = \lambda_i|\Delta\bar{u}_i|. \tag{8.106}$$

Let $|\rho| = \ell/\lambda$, from the constraint of $|\rho|$ we have $1 - \delta < \ell < 1 + \delta$. Substituting (8.106) into (8.104) yields

$$|\Delta\bar{u}_{i+1}| = |1 - |\rho|\lambda_i||\Delta\bar{u}_i| = |1 - \ell\frac{\lambda_i}{\lambda}||\Delta\bar{u}_i|.$$

The convergence of iteration learning is determined by the magnitude of the factor $|1 - \ell\frac{\lambda_i}{\lambda}|$. The upper bound for $|1 - \ell\frac{\lambda_i}{\lambda}|$ indicates the slowest convergence rate. Next we derive this upper bound with two scenarios.

Case 1. $\min\{\frac{\lambda}{\lambda_i}, 1 + \delta\} = \frac{\lambda}{\lambda_i}$. When $1 - \delta < \ell \leq \frac{\lambda}{\lambda_i}$,

$$|1 - \ell\frac{\lambda_i}{\lambda}| = 1 - \ell\frac{\lambda_i}{\lambda} < 1 - (1 - \delta)\frac{\lambda_i}{\lambda} \triangleq \delta_i < 1.$$

When $\frac{\lambda}{\lambda_i} < \ell < 1+\delta$,

$$\begin{aligned} |1-\ell\frac{\lambda_i}{\lambda}| &= \ell\frac{\lambda_i}{\lambda} - 1 < (1+\delta)\frac{\lambda_i}{\lambda} - 1 \\ &\leq \delta = 1-(1-\delta) \leq \delta_i. \end{aligned}$$

From $(1+\delta)\frac{\lambda_i}{\lambda} \leq 1+\delta$ we conclude $(1+\delta)\frac{\lambda_i}{\lambda} - 1 \leq \delta$ and, thus,

$$(1+\delta)\frac{\lambda_i}{\lambda} - 1 \leq \delta = 1-(1-\delta) \leq \delta_i.$$

Case 2. $\min\{\frac{\lambda}{\lambda_i}, 1+\delta\} = 1+\delta$. In this case, we have

$$|1-\ell\frac{\lambda_i}{\lambda}| = 1-\ell\frac{\lambda_i}{\lambda} < 1-(1-\delta)\frac{\lambda_i}{\lambda} = \delta_i.$$

Thus the upper bound of the convergence factor is

$$\delta_i = 1-(1-\delta)\frac{\lambda_i}{\lambda}. \tag{8.107}$$

for all iterations. Note that when $\bar{u}_i \neq \bar{u}^*$, $\bar{y}_i \neq y^*$ by Assumption 6.3, consequently $\lambda_i \neq 0$ by (8.106) and the upper bound ρ_i will be strictly less than 1 as far as $\bar{u}_i$ does not converge to $\bar{u}^*$.

Let ϵ_1 denote the desired ϵ_1-precision bound of learning, i.e. $|\Delta\bar{y}_i| < \epsilon_1$. Now we show that the sequence $\bar{y}_i$ can enter the prespecified ϵ_1-precision bound after a finite number of iterations.

First, considering the fact $\delta_i \leq 1$, using (8.107) repeatedly yields

$$|\Delta\bar{y}_i| = \lambda_i|\Delta\bar{u}_i| = \lambda_i\prod_{j=1}^{i-1}\delta_j|\Delta\bar{u}_0| \leq \lambda_i(v_{max}-v_{min}).$$

Before $\bar{y}_i$ enters the ϵ_1-bound,

$$\epsilon_1 < |\Delta\bar{y}_i| \leq \lambda_i|\Delta\bar{u}_0| \leq \lambda_i(v_{max}-v_{min})$$

which gives the lower bound of the coefficient λ_i, $\lambda_i \geq \epsilon_1/(v_{max} - v_{min})$ for all iterations before learning terminates. Similarly by using the relationship (8.107) repeatedly, and substituting the lower bound of λ_i, we can derive

$$\begin{aligned} |\Delta\bar{y}_i| &\leq \lambda|\Delta\bar{u}_i| \leq \lambda\prod_{j=1}^{i-1}\delta_j|\Delta\bar{u}_0| \\ &= \lambda\prod_{j=1}^{i-1}\left(1-(1-\delta)\frac{\lambda_j}{\lambda}\right)(v_{max}-v_{min}) \\ \leq\ & (v_{max}-v_{min})\lambda\left(1-(1-\delta)\frac{\epsilon_1}{(v_{max}-v_{min})\lambda}\right)^i \end{aligned}$$

which gives the upper bound of $|\Delta\bar{y}_i|$. Solving for $(v_{max}-v_{min})\lambda\left(1-(1-\delta)\frac{\epsilon_1}{(v_{max}-v_{min})\lambda}\right)^{i-1} \leq \epsilon_1$ with respect to i, the maximum number of iterations needed is

$$i \leq \frac{\log\frac{\epsilon_1}{(v_{max}-v_{min})\lambda}}{\log\left(1-(1-\delta)\frac{\epsilon_1}{(v_{max}-v_{min})\lambda}\right)} + 1.$$

Appendix B:
Published/Submitted Papers

Refereed Journal Articles:

[1] Jian-Xin Xu and Deqing Huang, "Spatial periodic adaptive control for rotary machine systems", *IEEE Trans. Automat. Control*, Vol. 53, No.10, pp.2402-2408, 2008.

[2] Jian-Xin Xu and Deqing Huang, "Initial state iterative learning for final state control in motion systems", *Automatica*, Vol. 44, pp.3162-3169, 2008.

[3] Jian-Xin Xu, Deqing Huang and Srinivas Pindi, "Optimal tuning of PID parameters using iterative learning approach", *SICE J. Control, Measurement, and System Integration*, Vol. 1, No. 2, pp.143-154, 2008.

[4] Yilei Tang, Deqing Huang, Shigui Ruan and Weinian Zhang, "Coexistence of limit cycles and homoclinic loops in a SIRS model with a nonlinear incidence rate", *SIAM J. Appl. Math.*, Vol. 69, pp.621-639, 2008.

[5] Deqing Huang and Jian-Xin Xu, "Iterative boundary learning control for a class of nonlinear PDE processes", Submitted.

[6] Deqing Huang and Jian-Xin Xu, "Discrete-time adaptive control for nonlinear systems with periodic parameters: A lifting approach", Submitted.

[7] Deqing Huang, Jian-Xin Xu, and Ying Tan, "A dual-loop iterative learning control for nonlinear systems with hysteresis input uncertainty", Submitted.

[8] Qingyi Guo, Deqing Huang, Chunlin Luo, and Weinian Zhang, "An infinite-dimensional global implicit function theorem and its applications to control of non-affine parabolic systems", Submitted.

International Conference Articles:

[1] Jian-Xin Xu, Wei Wang and Deqing Huang, "Iterative learning in ballistic control", In *Proc. IEEE American Control Conf.*, July, New York, USA, 2007.

[2] Jian-Xin Xu and Deqing Huang, "Optimal tuning of PID parameters using iterative learning approach", In *Proc. IEEE Multiple Conf. on Systems and Control*, October, Singapore, 2007.

[3] Jian-Xin Xu and Deqing Huang, "Spatial periodic adaptive control for rotary machine systems", In *Proc. Int. Federation of Automat. Control (IFAC)*, July, Korea, 2008.

[4] Jian-Xin Xu and Deqing Huang, "Initial state iterative learning for final state control in motion systems", In *Proc. Int. Federation of Automat. Control (IFAC)*, July, Korea, 2008.

[5] Deqing Huang, Jian-Xin Xu, and Zhongsheng Hou, "A discrete-time periodic adaptive control approach for parametric-strict-feedback systems", In *Proc. IEEE Conf. on Decision and Control*, December, Shanghai, 2009.

[6] Jian-Xin Xu and Deqing Huang, "Discrete-time adaptive control for nonlinear systems with periodic parameters: A lifting approach", In *Proc. 7th Asian Control Conf.*, August, Hong Kong, 2009.

[7] Deqing Huang, Ying Tan, and Jian-Xin Xu, "A dual-loop iterative learning control for nonlinear systems with hysteresis input uncertainty", In *Proc. 7th IEEE Int. Conf. on Control & Automation (ICCA09)*, December, New Zealand, 2009.

[8] Deqing Huang, Jian-Xin Xu, and Kai-Yew Lum, "Surveillance for a simply connected region: A one-center disk-covering problem", In *Proc. 8th IEEE Int. Conf. on Control and Automation*, Xiamen, China, 2010.

[9] Deqing Huang, Jian-Xin Xu, V. Venkataramanan, Weijie Lim, C. L. Eileen Chow, and T. C. Tuong Huynh, "Modeling and compensation of contact induced vibrations in high density hard disk drive servo systems", In *Proc. 36th Annual Conf. of the IEEE Industrial Electronics Society*, Glendale, AZ, USA, 2010.