Chapter 1
Introduction

1.1 Iterative Learning Control—Why and How

Learning is a basic skill of human being so that we can survive against the severe environments in the ancient times. Learning also helps us behave better and better in our daily lives and work nowadays. Through the learning process, we can improve our behavior by utilizing the experience from previous executions. Take the basketball shooting from a fixed point as an example. For the first shot, the shooter may fail to hit the basket and even the basketball may be far away from hitting. Nevertheless, the shooter would learn how the final behavior is and how he/she should do in the next shot. Thus, as the number of attempts increases, the shooter is able to increase the hit ratio since he/she may adjust the angle and speed to reduce the shooting deviation shot by shot. Such learning scenes are very common in our daily lives, for examples, we learn to speak, to write, and to drive, all by repeating. That is why we believe learning is a basic skill of human being.

Then, one is interested in whether such basic skill, i.e., learning, could be introduced to the design of automatic control so that the machine can own certain ability of learning. The answer is yes. As a matter of fact, this basic cognition motivates the introduction and developments of a novel learning control method, iterative learning control (ILC). Before proceeding to the formal formulation of ILC, let us first recall some observations on the learning process of human being. In order to learn from experience, repetition is a fundamental requirement of the process. That is, the learning objective should be kept the same in different iterations when we are learning; otherwise, the learning experience is difficult to be applied to a new task. Besides, a repeatable learning process provides us the space for asymptotical improvement so that we could continuously perfect our performance. In short, repetition is closely connected with learning.

When considering the control method where the idea of learning is involved, such repetition requirement is also reserved. As to ILC, it is designed for the systems that are able to complete certain task over a fixed time interval and perform them repeatedly. In such systems, the input and output information of past itera-

D. Shen, *Iterative Learning Control with Passive Incomplete Information*,
https://doi.org/10.1007/978-981-10-8267-2_1

tions/cycles/batches,[1] as well as the tracking objective, are used to formulate the input signal for the next iteration/cycle/batch, so that the tracking performance can be improved as the iteration number increases to infinity.

Thus, ILC has the following features: (1) the system can complete a task in a finite time interval, (2) the system can be reset to the same initial value, and (3) the tracking objective is iteration-invariant. We give a brief explanation for such features. First of all, the feature of finite operation length is a specified character for ILC, which is satisfied in many practical applications such as the production line automation and pick and place robot. If such feature is removed, i.e., the system would keep running along the time axis, certain learning mechanisms may be applicable if the system dynamics is periodic. Moreover, from the generic requirement of repetition, the resetting condition, i.e., the second feature, is a natural but specific formulation of ILC. Meanwhile, it is also a major criticizing point on ILC for practical applications. In the past three decades, such condition has attracted much attention and much literature has been published to relax it. Last but not least, the invariance of tracking objective is also a specific formulation of repetition. Noticing that the human being could apply the knowledge learned from other target to a new target but with similarities, we also expect such feature to be relaxed to the iteration-varying case. This issue has been addressed in many existing papers.

The main concept of ILC is shown in Fig. 1.1, where y_d denotes the reference trajectory. At the kth iteration, the input u_k is fed to the plant so that the corresponding output is y_k and the tracking error is $e_k = y_d - y_k$. This is a causal process. Since the tracking error e_k is nonzero, implying that the input u_k is not good enough, we will make an improvement to the input in the next iteration (i.e., the $(k+1)$th iteration). The specific construction for the improvement actually is a function of u_k and e_k, although it is usually specified as a linear combination. In other words, the input u_{k+1} for the $(k+1)$th iteration is generated before the running of the $(k+1)$th iteration. Then, the newly generated input is fed to the plant for the next iteration. Meanwhile, the input u_{k+1} is also stored in the memory for updating the input for the $(k+2)$th iteration. As a result, a closed-loop feedback is formed along the iteration axis.

By comparing ILC with our daily lives, we find that the previous information on inputs and outputs of the plant corresponds to the experiences accumulated in our daily lives. Persons usually decide a strategy for a given task based on previous experiences, while the strategy here corresponds to the input signal of ILC. Note that the previous experiences would help us to improve our behavior; thus, it is reasonable to believe that information on the previous operation may help improve the control performance to some extent.

The major advantage of ILC is that the design of control laws mainly requires the tracking references and input/output signals. In other words, little information about the plant is required and the system matrices may even be completely unknown. However, the algorithm is simple and effective.

---

[1] When we refer one operation process, the terminologies iteration, cycle, and batch are equivalent to each other.

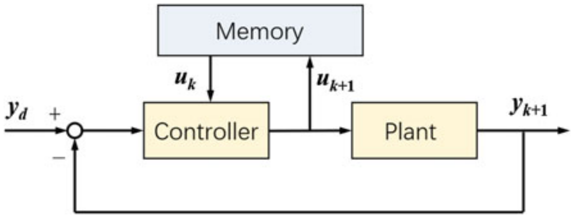

Fig. 1.1 Framework of ILC

It is important to note that ILC adjusts the control along the iteration index rather than the time index, which is the main difference with other control methods such as proportional–integral–derivative (PID) control. PID control is a widely used feedback control. However, for repetitive systems, PID generates the same tracking error for all iterations since no previous information is used, whereas ILC reduces the tracking error iteration by iteration. Additionally, ILC differs from adaptive control, which also learns from previous operation information. Adaptive control aims to adjust the parameter of a structure-fixed controller, while ILC aims to construct the input signal directly.

The concept of ILC may be traced back to a paper published in 1978 by Uchiyama [1]. However, this paper failed to attract widespread attention as it was written in Japanese. Three papers that were published in 1984 [2–4] resulted in the further research on ILC. Subsequently, large amounts of literature have been published on various related issues, such as research monographs [5–9], survey papers [10–12], and special issues of academic journals [13–16]. ILC has recently become an important branch of intelligent control, and its use is widespread in many practical applications such as robotics [17–20], hard disk drives [21, 22], and industrial processes [23, 24].

1.2 Basic Formulation of ILC

We now present a basic formulation of ILC, followed by some traditional convergence results. Note that computer has been widely applied in the control of practical systems, where the sampled control systems or equivalently discrete-time systems are investigated. Moreover, this monograph concentrates on the discrete-time system. Thus, we first present the discrete-time case and then give the continuous-time counterpart briefly.

1.2.1 Discrete-Time Case

Consider the following discrete-time linear time-invariant system:

$$\begin{aligned} x_k(t+1) &= Ax_k(t) + Bu_k(t), \\ y_k(t) &= Cx_k(t), \end{aligned} \tag{1.1}$$

where $x \in \mathbf{R}^n$, $u \in \mathbf{R}^p$, and $y \in \mathbf{R}^q$ denote the system state, input, and output, respectively. Matrices A, B, and C are system matrices with appropriate dimensions. t denotes an arbitrary time instant in an operation iteration, $t = 0, 1, \ldots, N$, where N is the length of the operation iteration. For simplicity, $t \in [0, N]$ is used in the following. $k = 0, 1, 2, \ldots$ denote different iterations.

Since it is required that a given tracking task should be repeated, the initial state needs to be reset at each iteration. The following is a basic reset condition, called identical initialization condition (i.i.c.), which has been used in many publications:

$$x_k(0) = x_0, \quad \forall k. \tag{1.2}$$

The reference trajectory is denoted by $y_d(t)$, $t \in [0, N]$. With regard to the reset condition, it is usually required that $y_d(0) = y_0 \triangleq Cx_0$. The control purpose of ILC is to design a proper update law for the input $u_k(t)$, so that the corresponding output $y_k(t)$ can track $y_d(t)$ as closely as possible. To this end, for any t in $[0, N]$, we define the tracking error as

$$e_k(t) = y_d(t) - y_k(t). \tag{1.3}$$

Then, the update law is a function of $u_k(t)$ and $e_k(t)$ to generate $u_{k+1}(t)$, whose general form is as follows:

$$u_{k+1}(t) = h(u_k(\cdot), \cdots, u_0(\cdot), e_k(\cdot), \cdots, e_0(\cdot)). \tag{1.4}$$

When the above relationship depends only on the last iteration, it is called first-order ILC update law; otherwise, it is called high-order ILC update law. Generally, to achieve the algorithm simplicity, most update laws are first-order laws, i.e.,

$$u_{k+1}(t) = h(u_k(\cdot), e_k(\cdot)). \tag{1.5}$$

Additionally, the update law is usually linear. The simplest update law is as follows:

$$u_{k+1}(t) = u_k(t) + Ke_k(t+1), \tag{1.6}$$

where K is the learning gain matrix, which is also the designed parameter. In (1.6), $u_k(t)$ is the input of the current iteration, while $Ke_k(t+1)$ is the innovation term. The update law (1.6) is called P-type ILC update law. If the innovation term is replaced by $K(e_k(t+1) - e_k(t))$, the update law is called D-type.

For system (1.1) and update law (1.6), a basic convergence condition is that K satisfies

$$\|I - CBK\| < 1. \tag{1.7}$$

Then, one has $\|e_k(t)\| \xrightarrow[k\to\infty]{} 0$, where $\|\cdot\|$ denotes the matrix or vector norm.

From this condition, one can deduce that the design of K needs no information with regard to the system matrix A, but requires information of the coupling matrix

CB only. This fact illustrates the advantage of ILC from the perspective that ILC has little dependence on the system information. Thus, ILC can handle tracking problems that have more uncertainties.

Remark 1.1 From the formulation of ILC, one can see that the model takes the classic features of a 2D system. That is, the system dynamics (1.1) and the update law (1.6) evolve along the time and iteration axes, respectively. Many researchers have made contributions from this point of view, and developed a 2D system-based approach, which is one of the principal techniques for ILC design and analysis.

Note that the operation length is limited by N and is then repeated multiple times. Thus, one could use the so-called lifting technique for discrete-time systems, which implies lifting all of the inputs and outputs as supervectors,

$$U_k = [u_k^T(0), u_k^T(1), \cdots, u_k^T(N-1)]^T, \tag{1.8}$$

$$Y_k = [y_k^T(1), y_k^T(2), \cdots, y_k^T(N)]^T. \tag{1.9}$$

Denote

$$G = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \cdots & \cdots & CB \end{bmatrix}, \tag{1.10}$$

then, one has

$$Y_k = GU_k + \mathbf{d}, \tag{1.11}$$

where

$$\mathbf{d} = [(CAx_0)^T, (CA^2x_0)^T, \ldots, (CA^Nx_0)^T]^T. \tag{1.12}$$

Similar to (1.8) and (1.9), define

$$Y_d = [y_d^T(1), y_d^T(2), \cdots, y_d^T(N)]^T,$$
$$E_k = [e_k^T(1), e_k^T(2), \cdots, e_k^T(N)]^T,$$

then it leads to

$$U_{k+1} = U_k + \mathbf{K}E_k, \tag{1.13}$$

where $\mathbf{K} = \text{diag}\{K, K, \cdots, K\}$. By simple calculation, one has

$$\begin{aligned} E_{k+1} =& Y_d - Y_{k+1} = Y_d - GU_{k+1} - \mathbf{d} \\ =& Y_d - GU_k - G\mathbf{K}E_k - \mathbf{d} \\ =& E_k - G\mathbf{K}E_k \\ =& (I - G\mathbf{K})E_k. \end{aligned}$$

Therefore, we obtain a condition (1.7) that is sufficient to guarantee the convergence of ILC. Actually, the lifting technique not only helps us to obtain the convergence condition but also provides us with an intrinsic understanding of ILC. In the lifted model (1.11), the evolutionary process of an operation iteration has been integrated into G, whereas the relationship between adjacent iterations is highlighted. That is, the lifted model (1.11) depends on the k-axis only.

Remark 1.2 Note that the focus of ILC is how to improve the tracking performance gradually along the iteration axis, as one can see from the design of the update law (1.13) and lifted model (1.11). Therefore, it would not cause additional difficulties when the system is extended from the linear time-invariant case to the linear time-varying case. This is because, for any fixed time, the updating process along the iteration axis is a time-invariant case.

It is usually assumed that the reference trajectory $y_d(t)$ is realizable. That is, there exists an appropriate initial state x_0 and input $u_d(t)$ such that the expression (1.1) still holds with k replaced by d. In other words, $Y_d = GU_d + \mathbf{d}$, where U_d is defined in a similar manner of (1.8). Then, the discussion that the system output converges to the reference trajectory, i.e., $\lim_{k\to\infty} Y_k = Y_d$, becomes one that the system input converges to the objective input, i.e., $\lim_{k\to\infty} U_k = U_d$. For the system with stochastic noises, this transformation is more convenient for convergence analysis.

Remark 1.3 One may be interested in the case that the reference trajectory is not realizable. In other words, there is no control input producing the reference trajectory; thus, entirely accurate tracking is impossible. Then, the design objective of the ILC algorithm is no longer to guarantee asymptotically accurate tracking, but to converge to the nearest trajectory of the given reference. Consequently, the tracking problem has become an optimization problem. On the other hand, from the point of view of practical applications, the reference trajectory is usually realizable; thus, the assumption is not rigorous.

1.2.2 Continuous-Time Case

Let us consider the following linear continuous-time system:

$$\begin{aligned} \dot{x}_k(t) &= Ax_k(t) + Bu_k(t), \\ y_k(t) &= Cx_k(t), \end{aligned} \tag{1.14}$$

where the notations have similar meaning to the discrete-time formulation.

The control task is to servo the output y_k to track the desired reference y_d on a fixed time interval $t \in [0, T]$ as the iteration number k increases. If the system has relative degree one, an ILC scheme of Arimoto-type can be given as

$$u_{k+1} = u_k + \Gamma \dot{e}_k, \tag{1.15}$$

where $e_k(t) = y_d(t) - y_k(t)$ and Γ is the diagonal learning gain matrix. Similarly, if the learning gain matrix satisfies

$$\|I - CB\Gamma\| < 1, \tag{1.16}$$

then the control purpose is ensured, i.e., $\lim_{k\to\infty} y_k(t) \to y_d(t)$. Note that the basic formula for selecting the learning gain matrix given in (1.16) requires no information about the system matrix A, which implies that ILC can be effective for model-uncertain systems.

Moreover, a "PID-like" update law can be formulated as

$$u_{k+1} = u_k + \Phi e_k + \Gamma \dot{e}_k + \Psi \int e_k \mathrm{d}t, \tag{1.17}$$

where Φ, Γ, and Ψ are learning gain matrices. The high-order PID-like update law can be formulated as

$$u_{k+1} = \sum_{i=1}^{n} (I - \Lambda) P_i u_{k+1-i} + \Lambda u_0 + \sum_{i=1}^{n} \left(\Phi_i e_{k+1-i} + \Gamma_i \dot{e}_{k+1-i} + \Psi_i \int e_{k+1-i} \mathrm{d}t \right), \tag{1.18}$$

where $\sum_{i=1}^{n} P_i = I$.

1.3 ILC with Random Data Dropouts

We first provide three common models of data dropouts and detail the differences among the models. Throughout this monograph, the data dropout occurring or not can be regarded as a switch that opens and closes the network in a random manner. It is denoted by a random variable $\gamma_k(t)$. Therefore, there are two possible states of the variable $\gamma_k(t)$. Specifically, we let $\gamma_k(t)$ be equal to 1 if the corresponding data packet is successfully transmitted, and let $\gamma_k(t)$ be equal to 0 otherwise.

Generally, we have the following three most common models of data dropouts.

- Random sequence model (RSM): For each t, the measurement packet loss is random without obeying any certain probability distribution, but there is a positive integer $K \geq 1$ such that at least in one iteration the measurement is successfully sent back during the successive K iterations.
- Bernoulli variable model (BVM): The random variable $\gamma_k(t)$ is independent for different time instant t and iteration number k. Moreover, $\gamma_k(t)$ obeys a Bernoulli distribution with

$$\mathbb{P}(\gamma_k(t) = 1) = \overline{\gamma}, \quad \mathbb{P}(\gamma_k(t) = 0) = 1 - \overline{\gamma}, \tag{1.19}$$

  where $\overline{\gamma} = \mathbb{E}\gamma_k(t)$ with $0 < \overline{\gamma} < 1$.

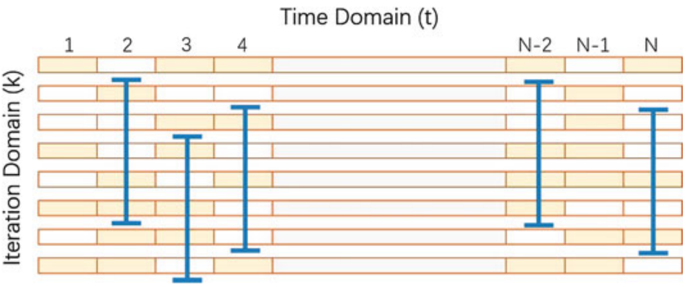

Fig. 1.2 Illustration of the RSM

- Markov chain model (MCM): The random variable $\gamma_k(t)$ is independent for different time instant t. Moreover, for each t, the evolution of $\gamma_k(t)$ along the iteration axis follows a two-state Markov chain, of which the probability transition matrix is

$$P = \begin{bmatrix} P_{11} & P_{10} \\ P_{01} & P_{00} \end{bmatrix} = \begin{bmatrix} \mu & 1-\mu \\ 1-\nu & \nu \end{bmatrix} \tag{1.20}$$

  with $0 < \mu,\ \nu < 1$, where $P_{11} = \mathbb{P}(\gamma_{k+1}(t) = 1 \mid \gamma_k(t) = 1)$, $P_{10} = \mathbb{P}(\gamma_{k+1}(t) = 0 \mid \gamma_k(t) = 1)$, $P_{01} = \mathbb{P}(\gamma_{k+1}(t) = 1 \mid \gamma_k(t) = 0)$, $P_{00} = \mathbb{P}(\gamma_{k+1}(t) = 0 \mid \gamma_k(t) = 0)$.

Remark 1.4 The RSM is illustrated in Fig. 1.2 where a horizontal bar denotes an iteration process. In any bar, the white rectangle and light gold rectangle denote the lost packet and successfully transmitted packet, respectively. The gray part of each horizontal bar denotes the omission part. The RSM implies that, for arbitrary time instant t, the corresponding output information can be received at least once for any successive K iterations. As shown in Fig. 1.2, we take $K = 5$ for example. It is seen that there is at least one colored rectangle for any successive K horizontal bars and for any time instant. Moreover, this model can be formulated using the random variable $\gamma_k(t)$ as follows: for each t, $\sum_{i=0}^{K-1} \gamma_{k+i}(t) \geq 1$ for all $k \geq 1$. It is worth pointing out that we only require the existence of the number K rather than its specific value; that is, the number K is not necessary to be known prior and it is not involved in the design of ILC update law later. In fact, this model means that the output information should not be lost too much to ensure the learning ability in a somewhat deterministic point of view.

Remark 1.5 The number K of the RSM indicates that the maximum length of successive data dropouts is $K - 1$. Thus, the case $K = 1$ means no data dropout occurring, while the case $K = 2$ means no successive data dropout occurring for any two subsequent iterations. Moreover, the value of the successive iteration number K is a reflection of the rate of data dropouts. However, it is not equivalent to the data dropout rate (DDR), which can be formulated as $\lim_{n\to\infty} 1/n \times \left[\sum_{k=1}^{n} \left(1 - \gamma_k(t)\right)\right]$.

In fact, DDR denotes the average level of data dropouts along the iteration axis, while K implies the worst case of successive data dropouts. In other words, a larger K usually corresponds to a higher DDR, while a smaller K usually corresponds to a lower DDR. But the connection between K and DDR is not necessarily to be positively related.

Remark 1.6 The mathematical expectation $\overline{\gamma}$ of the BVM is closely related to the DDR in the light of the law of large numbers; that is, DDR is equal to $1-\overline{\gamma}$. Specifically, the data dropout is independent along the iteration axis, thus, $\lim_{n\to\infty} 1/n \times \left[\sum_{k=1}^{n}\left(1-\gamma_k(t)\right)\right] = 1-\mathbb{E}\gamma_k(t) = 1-\overline{\gamma}$. If $\overline{\gamma}=0$, implying that the network is completely broken down, then no information can be received from the plant, and thus no algorithm can be applied to improve the tracking performance. If $\overline{\gamma}=1$, implying that no data dropout occurs, then the framework converts into the classical ILC problem. In this monograph, we simply assume $0<\overline{\gamma}<1$. Moreover, the statistics property of $\gamma_k(t)$ is assumed to be identical for different time instants for concise expression. The extension to the time-dependent case, i.e., the case that $\mathbb{E}\gamma_k(t)=\overline{\gamma}_t$, is straightforward without additional efforts.

Remark 1.7 The MCM is general for modeling the data dropouts. The transition probabilities μ and ν denote average levels of retaining the same state for successful transmission and loss, respectively. If $\mu+\nu=1$, then MCM converts into BVM. That is, BVM is a special case of MCM. It is worth pointing out that all the three models are widely investigated in the field of networked control systems, such as [25] for RSM, [26] for BVM, and [27] for MCM.

Remark 1.8 In this remark, we comment the differences among the three models. RSM differs from both BVM and MCM as it requires no probability distribution or statistics property of the random variable $\gamma_k(t)$. However, RSM pays the price that the successive data dropouts length is bounded comparing with BVM and MCM. Specifically, both BVM and MCM admit arbitrary successive data dropouts associated with a suitable occurring probability. Consequently, RSM cannot cover BVM/MCM and vice versa. It should be pointed out that RSM implies that the data dropout is not totally stochastic. Moreover, the difference between BVM and MCM lies in the point that the data dropout occurs independently along the iteration axis for BVM, while dependently for MCM. The independence of data dropout admits some specific computations such as mean and variance, and then derives the convergence analysis. Such technique is not applicable for MCM.

Next, we give a brief literature review on ILC for systems with random data dropouts and classify the contributions of the existing papers from three aspects, namely, random data dropout models, data dropouts positions, and the convergence meaning. From these three dimensions, we can get a comprehensive picture view of the state of the art.

1.3.1 Data Dropout Models

The most popular model for the data dropout should go to BVM. In this model, the random variable takes the value 1 with success probability $\overline{\gamma}$ and the value 0 with failure probability $1-\overline{\gamma}$. Besides, the data dropouts for different packets occur independently. In other words, this model has a clear probability distribution and good independence. Therefore, it is widely used in many existing papers addressing data dropout topic. Most ILC papers adopt this model [28–41] with/without extra requirements on data dropouts.

There are a few ILC papers dropping this model. In [42], the authors contributed to give an elaborate investigation of the effect of data dropouts. Thus, the authors mainly considered the case that only a single packet was lost during the transmission and provided a specific derivation for the effect on the input error and tracking performance. As to the multiple packet loss case, a general discussion was given instead of strict analysis and description. Specifically, the authors claimed that the data dropout level should be far smaller than 100% to ensure a satisfied tracking performance.

The papers [43, 44] provided RSM for data dropouts. Specifically, the sequence of the data dropout variables along the iteration axis was not assumed to be with any specific probability distribution. In other words, the statistical property of the data dropouts can vary along the iteration axis. Thus, the steady distribution in the Bernoulli model is removed. However, in order to ensure an asymptotical convergence of the input sequence, an additional requirement was imposed to the data dropout model in [43, 44] that there should exist a sufficiently large number K such that during any successive K iterations, at least one data dropout variable takes value 1. In other words, the data should be successfully transmitted from time to time.

There is another model for data dropouts, MCM, which has been used in some papers addressing other control strategies. In this model, the data dropouts have some dependence on the previous event. That is, the loss or not of the current packet would affect the probability of successful transmission for the next packet. In the ILC under data dropouts, this model has been discussed in very few papers.

1.3.2 Data Dropout Positions

In the networked ILC, the plant and the learning controller are separated in different sites and communicate with each other through wired/wireless networks. Thus, there are two channels connecting the plant and the learning controller. One is at the measurement side to transmit the measured output information back to the learning controller. The other one is at the actuator side to transmit the generated input signal to the plant so that the operation process can continuously run.

When considering the data dropouts problem for ILC, the position at which data dropout occurs is usually assumed to be the measurement side. In other words, only the network at the measurement side is assumed to be lossy and the network at the actuator side is assumed to work well in most papers such as [28–31, 33, 34, 37–40, 43, 44]. In these papers, the generated input signal can be always sent to the plant without any loss. Although some papers claimed that their results can be extended to the general case where both the networks at the measurement and actuator sides suffered random data dropouts, it is actually not a trivial extension.

Specifically, when the network at the measurement side suffers random data dropouts, the output signal of the plant may or may not be successfully transmitted. One simple mechanism for treating the measured data is as follows: if the measured output is successfully transmitted, then the learning controller would employ such information for updating; if the measured output is lost during the transmission, then the learning controller would stop updating until the corresponding output information is successfully transmitted. One may find that the lost data is simply replaced by 0 in this mechanism. For the case that data dropout occurs only at the measurement side, such simple mechanism is sufficient to ensure the learning process as long as the network is not completely broken down. However, when considering the data dropout at the actuator side, it is clear that the lost input signal cannot be simply replaced by 0 as it would greatly damage the tracking performance. That is, if the network at the actuator side suffers data dropouts, the lost input signal must be compensated with a suitable packet to maintain the operation process of the plant. This observation motivates the investigation on compensation mechanisms for the lost data [32, 35, 36, 42].

In [42], the authors gave an earlier attempt on compensating the lost data. When one packet of the input signal is lost at the actuator side, the one-time-instant ahead input signal is applied to compensate the lost one. That is, if the input at time instant t is lost, it would be compensated with the input at time instant $t - 1$. When one packet of the output signal is lost at the measurement side, a similar compensation mechanism is applied. It is worth noting that the data dropouts at the measurement side and actuator side are separately discussed in [42]. Moreover, this mechanism was then adopted by [32] for a Bernoulli model of random data dropouts occurring at both the measurement and actuator sides simultaneously. We should emphasize that, as a natural consequence, the data at adjacent time instants of the same iteration cannot be dropped simultaneously due to the inherent compensation requirement. Another compensation mechanism is to apply the corresponding data from the last iteration as shown in [35, 36]. That is, if the data packet at the kth iteration is lost during the transmission, it is compensated with the packet at the $(k - 1)$th iteration with the same time instant label. In such assumption, the successive data dropouts along the time axis are allowed; however, it restricts that there was no simultaneous data dropout at the same time instant across any two adjacent iterations. In other words, no successive data dropouts along the iteration axis are allowed.

In short, the contributions in [32, 35, 36, 42] show that the newly introduced compensation mechanisms impose additional limitations on the data dropout models. In fact, the inherent difficulty of convergence analysis lies in the asynchronism between the computed input of the learning controller and the actual input fed to the plant. A recent paper [41] solved this problem according to the Bernoulli model allowing successive data dropouts along both time and iteration axes and provided a simple compensation mechanism with the iteration-latest available packet.

1.3.3 Convergence Meanings

In this subsection, we review the analysis techniques and the related convergence results, especially the convergence meanings in consideration of the randomness of data dropouts besides optional stochastic noises.

Ahn et al. provided earlier attempts to the ILC for linear systems in the presence of data dropouts [28–30]. The Kalman filtering based technique, which was first proposed by Saab in [45], was applied and thus the mean square convergence of the input sequence was obtained. The main difference among the papers [28–30] lies in the position where data dropouts occur. Specifically, in the first paper [28], the output vector was assumed to be lossy, and in [29] this assumption was relaxed to the case only partial dimensions of the output may suffer data dropouts. Last, in [30], the data dropouts at both the measurement and actuator sides were taken into account. In short, the Kalman filtering based technique was deeply investigated in these papers.

Bu et al. gave some different angles to solve this problem in [31–34]. In [31], the exponential stable result of asynchronous dynamical systems [46] was referred to establish the convergence condition of ILC under data dropouts. As a result, the randomness of data dropouts was not involved in the analysis steps. In [32], such randomness was eliminated from the recursion by taking mathematical expectation, thus the algorithm was converted into deterministic type and then the design and analysis of the convergence followed the conventional way. Therefore, the convergence was clear in the mathematical expectation sense. In [33], a new H-∞ framework was defined along the iteration axis and then the related control problem was solved in the newly defined framework. That is, the kernel objective was to satisfy an H-∞ performance index in the mean square sense. An LMI design condition for the learning gain matrix was also provided. In [34], the widely used 2D system approach was revisited to deal with data dropouts. A mean square asymptotically stable result was obtained and the design condition for the learning gain matrix was solved through LMI techniques. In short, the evolution dynamics along the iteration axis was carefully studied and related techniques are applied for the design and analysis of ILC.

There are several scattered results on this topic including [35–37, 42]. The paper [42] proposed a detailed analysis of the effect of packet loss for the sampled ILC. Specifically, a single packet loss at the measurement side and the actuator side was evaluated separately to study the inherent influence of data dropout on the tracking performance. In other words, a deterministic analysis was given according to the input error. The results in [42] revealed that neither contraction nor expansion occurred for the input error if the corresponding packet was lost during the transmission. Such technique was further exploited in [35] to study the general data dropout case.

In [36], a mathematical expectation was taken to the recursive inequality of input error to eliminate the randomness of data dropouts similar to [32] and then the conventional contraction mapping method was used to derive the convergence results. Moreover, to construct an explicit contraction mapping, the conditions in [36] were much conservative and it may be further relaxed. Similar techniques were also used in [37] incorporated with the conventional λ-norm technique to derive a convergence in mathematical expectation sense.

We mainly contribute the almost sure convergence results of ILC under data dropouts environments. In [38], a simple case that the whole iteration was packed and transmitted as a single packet was investigated by a switched system approach. Specifically, the evolution along the iteration axis was formulated as a switched system and the statistical properties were recursively computed. Then, the convergence in the sense of expectation, mean square, and almost sure was established in turn. In [43], based on stochastic approximation theory, the almost sure convergence of the input sequence was proved for the case that the data dropouts were modeled by a random sequence. This result was then extended to the unknown control direction case in [44]. For the traditional Bernoulli model of data dropouts, the essential difficulty in obtaining the almost sure convergence lies in the random successive data dropouts along the iteration axis. This problem was solved in [39] and [40] for linear and nonlinear stochastic systems, respectively. We then proceeded to investigate the general data dropouts at both measurement and actuator sides without any additional requirements except the Bernoulli assumption in [41]. When data dropouts occur at the actuator sides, there is a newly introduced asynchronism between the computed control generated by the learning controller and the actual control fed to the plant. Such asynchronism was characterized by a Markov chain in [41] and then the mean square and almost sure convergence were established.

The recent progress on ILC in the presence of data dropouts is classified in Table 1.1 according to data dropout model, data dropout position, and convergence meaning. From this table, we observe the following points:

- In most papers, the data dropout is modeled by the Bernoulli random variable, whereas the results according to random sequence model are rather limited. Moreover, for the Markov chain model, few results have been reported.
- All the papers consider the data dropout occurring at the measurement side and only a few papers address the case at the actuator side. As we have explained above, the latter case would involve an essential influence on the controller design and convergence analysis.
- The convergence meanings are scattered in different papers. Both mean square and almost sure convergence imply the convergence of mathematical expectation sense. However, they cannot imply each other according to the probability theory. Thus, it is of interest to propose an in-depth framework for the design and analysis of ILC in both senses simultaneously.

Table 1.1 Classification of the papers on ILC under data dropouts

Refs.	Model			Position		Convergence			
	RSM	BVM	MCM	Measurement	Actuator	M.E.	M.S.	A.S.	D.A.
[28]		•		•			•		
[29]		•		•			•		
[30]		•		•	•		•		
[31]		•		•					•
[32]		•		•	•	•			
[33]		•		•			•		
[34]		•		•			•		
[35]		•		•	•				•
[36]		•		•	•	•			
[37]		•		•		•			
[38]		•		•		•	•	•	
[39]		•		•				•	
[40]		•		•				•	
[41]		•		•	•		•	•	
[42]		•		•	•				•
[43]	•			•				•	
[44]	•			•				•	

RSM: random sequence model, BVM: Bernoulli variable model, MCM: Markov chain model, M.E.: mathematical expectation, M.S.: mean square, A.S.: almost sure, D.A.: deterministic analysis

1.4 ILC with Other Incomplete Information

1.4.1 Communication Delay and Asynchronism

In ILC, very few papers have been published on the issue of communication delay or communication asynchronism. A recent paper [36] considered this problem for discrete-time systems, where a P-type networked ILC scheme was proposed. In the scheme, the delayed data was compensated by the data from the previous iteration, and as a result, successive delays along the iteration axis were not allowed. Based on such assumption of the communication delay, a deterministic convergence analysis was given in the paper following the conventional contraction mapping principle. Indeed, the problem formulated in [36] can be transformed as an ILC problem with data dropout, which has been briefed in the last section.

A general model of communication delay or communication asynchronism was addressed in [47] for a large-scale system consisting of several subsystems, where the communication among different subsystems suffered random and possibly asynchronous communication delays. In particular, for large-scale systems, it is hard for each subsystem to learn all information of the whole system when generating control signals; hence, decentralized control is more suitable. Due to potentially different working efficiency values among subsystems, the control action may not be updated for all subsystems at each iteration step. Therefore, the update of all subsystems would introduce random asynchronism. We note that such problem can be solved following a similar method for the data dropout problem described by a random sequence model.

Time delay was widely investigated as it may reduce the performance of systems, which has been reported in many traditional control methodologies. In ILC, some results can be found in [48, 49]. However, the essence of ILC is to adjust the input signal using the input and output information of previous iterations; thus, the repetitive information along the iteration axis would not significantly affect ILC. As is well known, one of the main advantages of ILC is its reduced dependence on system information; thus, unknown but fixed time delays would have no impact on control performance since it could be regarded as a part of the information of the system. This intuition was verified in [50], where a class of affine nonlinear systems with time delays were studied. Generally, we believe that the random time delay would make a significant effect on system performance. However, there is yet no complete or explicit reply to this question. Thus, further explorations should be made regarding the basic influence of random time delays on ILC performance.

1.4.2 Iteration-Varying Lengths

In many practical applications, the inherent iteration-invariance is often violated due to unknown uncertainties or unpredictable factors. In this subsection, we concentrate on necessity of identical trial lengths, which have been found to be invalid in certain biomedical application systems. For example, while applying ILC in a functional electrical stimulation for upper limb movement and gait assistance, it has been seen that the operation processes end early for at least the first few passes due to safety considerations that the output may significantly deviate from the desired trajectory [51]. The associated variable-trial-length problem is detailed in [52] and [53], which clearly demonstrates the violation of the identical-trial-length assumption typically used in ILC.

There were some early research attempts to provide a suitable design and analysis framework for iteration-varying-length ILC that formed the groundwork for subsequent investigations [51–53]. For example, based on the experimental verifications and primary analysis of the convergence property given in [51–53], a systematic proof of the monotonic convergence in different norm senses was elaborated in [54] for linear systems. In particular, the necessary and sufficient conditions for monotonic convergence were discussed, as well as other issues including the controller design guidelines and influence of disturbances. Most importantly, no specific formulation of varying length was imposed in this framework. The first random model of varying-length iterations was proposed in [55] for discrete-time systems, and it was then extended to continuous-time systems in [56]. In these models, a stochastic variable was used to represent the occurrence of the output at each time instant and iteration, and it was then multiplied with the tracking error, which denoted the actual information of the updating process. To compensate for the information loss caused by randomly varying trial lengths, an iteration-average operator of all historical data was introduced to the ILC algorithm in [55], whereas in [56], this average operator was replaced by an iteration-moving-average operator to reduce the influence of very old data. Moreover, a lifted framework of ILC for a discrete-time linear system was provided in [57] to avoid the conservatism of the conventional λ-norm-based analysis in [55, 56]. We note that all of the contributions in [55–57] have two disadvantages: (1) they only obtained asymptotical convergence with respect to the expected value, which is a very weak convergence criteria for the control of stochastic models, and (2) the distribution of the introduced stochastic variable should be known to the controller. Therefore, one would wonder whether stronger convergence could be obtained. The answer to this question was given in [58] for a simpler ILC update law. Specifically, the discrete-time linear system was revisited, and the traditional P-type ILC law was employed. The authors transformed the error evolution along the iteration axis by modeling it as a switching system and then established the statistical properties of input errors (i.e., the mathematical expectations and covariances) in a recursive form. The convergence in the mathematical expectation, mean square, and almost sure senses was derived simultaneously. The results were then extended to a class of affine nonlinear systems in [59] using different analysis techniques. A recent work [60] further proposed two novel and improved ILC schemes based on the iteration-moving-average operator, in which a searching mechanism was additionally introduced to collect useful information while avoiding redundant tracking information from the past. We should note that, as opposed to [55–57], no specific distribution of the stochastic variable was required in these works.

In addition, some extensions have also been reported in the existing literature. In particular, nonlinear stochastic systems were taken into account in [61], and the bounded disturbances were included. In that study, the average-operator-based scheme was improved by collecting all available information. Nevertheless, we note that a Gaussian distribution of the variable pass length was required, which limits the possible application range. In [62], the authors extended the method to discrete-time linear systems with a vector relative degree. In this case, one needs to carefully select the output data for the learning algorithms to function. Additionally, this issue was further extended to stochastic impulse differential equations in [63] and fractional order systems in [64]. We would like to note that the convergence analyses derived in these papers are primarily based on the mature contraction mapping method, and thus are similar to [55].

1.5 Structure of This Monograph

In this monograph, we concentrate on ILC with passive incomplete information. Hereafter, by passive incomplete information, we mean the incomplete information and data caused by practical system limitation during data collecting, storing, trans-

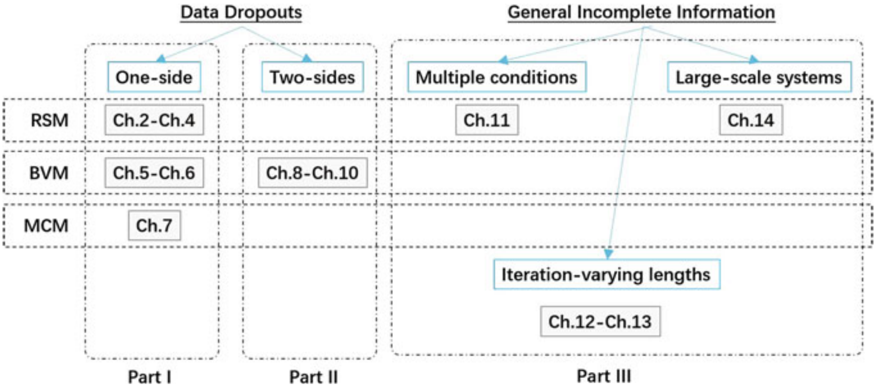

Fig. 1.3 Structure of this monograph

mitting, and processing stages, such as data dropouts, delays, disordering, and limited transmission bandwidth. Our primary objective is to design and analyze the corresponding ILC algorithms. In other words, we focus on two aspects, one of which is how to design the data compensation mechanism when certain data is lost and then design the ILC scheme, and the other one is to establish a unified framework for convergence analysis in both almost sure and mean square senses. The investigations in this monograph would help to understand the restrictive relationship between incomplete information and tracking performance quantitatively.

The rest of this monograph consists of three parts, as shown in Fig. 1.3. Part I aims to provide in-depth discussions on ILC under various data dropout models, where the data dropout is assumed to occur at the measurement side only. In particular, three models, i.e., RSM, BVM, and MCM, are all addressed in detail. This part contains six chapters with three chapters on RSM case, two chapters on BVM case, and one chapter on MCM case. Part II devotes to the general data dropout environments that the data dropout occurs at both measurement and actuator sides. This part contains three chapters elaborating linear deterministic systems, linear stochastic systems, and nonlinear systems in sequence. Part III provides more results on ILC with other types of passive incomplete information such as communication delay and iteration-varying lengths. Four chapters are included in this part concerning on multiple communication conditions, iteration-varying lengths, and asynchronism among subsystems of a large-scale system.

The chapters in this monograph can also be viewed from another angle, i.e., the random model angle (see Fig. 1.3). In particular, the convergence analysis was conducted based on RSM in Chaps. 2–4, 11, and 14, and based on BVM in Chaps. 5–10. We note that the model for iteration-varying lengths in Chaps. 12 and 13 can be regarded as a generalized Bernoulli model, and thus the techniques in Chaps. 12 and 13 can be applied to solve the BVM-based problem.

1.6 Summary

In this chapter, we present the fundamental principles of ILC first to clarify why and how we design ILC for repetitive systems. Then, we proceed to formulate the basic framework of ILC in both discrete-time and continuous-time forms. The literature on ILC with various types of passive incomplete information is reviewed and analyzed. Lastly, we provide a visual structure of the whole monograph.

References

1. Uchiyama, M.: Formulation of high-speed motion pattern of a mechanical arm by trial. Trans. SICE(Soc. Instrum. Contr. Eng.) **14**(6), 706–712 (1978)
2. Arimoto, S., Kawamura, S., Miyazaki, F.: Bettering operation of robots by learning. J. Robotic Syst. **1**(2), 123–140 (1984)
3. Casalino, G., Bartolini, G.: A learning procedure for the control of movements of robotic manipulators. In: Proceedings of the IASTED Symposium Robotics and Automation, pp. 108–111 (1984)
4. Craig, J.: Adaptive control of manipulators through repeated trials. In: Proceedings of the American Control Conference, pp. 1566–1573 (1984)
5. Moore, K.L.: Iterative Learning Control Control for Deterministic Systems. Springer, Berlin (1993)
6. Bien, Z., Xu, J.-X.: Iterative Learning Control - Analysis, Design, Integration and Applications. Kluwer Academic Publishers, Dordrecht (1998)
7. Chen, Y.Q., Wen, C.: Iterative Learning Control: Convergence, Robustness and Applications. Springer, London (1999)
8. Xu, J.-X., Tan, Y.: Linear and Nonlinear Iterative Learning Control. Springer, New York (2003)
9. Ahn, H.-S., Moore, K.L., Chen, Y.Q.: Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems. Springer, Berlin (2007)
10. Bristow, D.A., Tharayil, M., Alleyne, A.G.: A survey of iterative learning control: a learning-based method for high-performance tracking control. IEEE Control Syst. Mag. **26**(3), 96–114 (2006)
11. Ahn, H.-S., Chen, Y.Q., Moore, K.L.: Iterative learning control: survey and categorization from 1998 to 2004. IEEE Trans. Syst. Man Cybern. Part C **37**(6), 1099–1121 (2007)
12. Wang, Y., Gao, F., Doyle III, F.J.: Survey on iterative learning control, repetitive control and run-to-run control. J. Process Control **19**(10), 1589–1600 (2009)
13. Moore, K.L., Xu, J.-X.(Guest eds.): Special issue on iterative learning control. Int. J. Control **73**(10), 819–999 (2000)
14. Special issue on iterative learning control. Asian J. Control **4**(1), 1–118 (2002)
15. Ahn, H.-S., Moore, K.L.(Guest eds.): Special issue on iterative learning control. Asian J. Control **13**(1), 1–212 (2011)
16. Freeman, C.T., Tan, Y.(Guest eds): Special issue on iterative learning control and repetitive control. Int. J. Control **84**(7), 1193–1294 (2011)
17. Tayebi, A., Abdul, S., Zaremba, M.B., Ye, Y.: Robust iterative learning control design: application to a robot manipulator. IEEE/ASME Trans. Mechatron **13**(5), 608–613 (2008)
18. Freeman, C., Lewin, P., Rogers, E., Ratcliffe, J.: Iterative learning control applied to a gantry robot and conveyor system. Trans. Inst. Meas. Control **32**(3), 251–264 (2010)
19. Inaba, K.: Iterative learning control for industrial robots with end effector sensing. Ph.D. dissertation, University of California, Berkeley (2008)

20. Hoelzle, D.J., Alleyne, A.G., Johnson, A.J.W.: Iterative Learning Control for Robotic Deposition Using Machine Vision. In: Proceedings of the American Control Conference, pp. 4541–4547 (2008)
21. Chen, Y.Q., Moore, K.L., Yu, J., Zhang, T.: Iterative learning control and repetitive control in hard disk drive industry–a tutorial. Int. J. Adap. Control Signal Process. **22**(4), 325–343 (2008)
22. Wu, S.-C., Tomizuka, M.: An iterative learning control design for self-servoWriting in hard disk drives. Mechatronics **20**(1), 53–58 (2010)
23. Liu, T., Gao, F.: IMC-based iterative learning control for batch processes with time delay variation. J. Process Control **20**(2), 173–180 (2010)
24. Liu, T., Gao, F.: Robust two-dimensional iterative learning control for batch processes with state delay and time-varying uncertainties. Chem. Eng. Sci. **65**(23), 6134–6144 (2010)
25. Lin, H., Antsaklis, P.J.: Stability and persistent disturbance attenuation properties for networked control systems: switched system approach. Int. J. Control **78**(18), 1447–1458 (2005)
26. Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M.I., Sastry, S.S.: Kalman filtering with intermittent observations. IEEE Trans. Autom. Control **49**(9), 1453–1464 (2004)
27. Shi, Y., Yu, B.: Output feedback stabilization of networked control systems with random delays modeled by Markov chains. IEEE Trans. Autom. Control **54**(7), 1668–1674 (2009)
28. Ahn, H.S., Chen, Y.Q., Moore, K.L.: Intermittent iterative learning control. In: Proceedings of the 2006 IEEE International Symposium on Intelligent Control, pp. 832–837 (2006)
29. Ahn, H.S., Moore, K.L., Chen, Y.Q.: Discrete-time intermittent iterative learning controller with independent data dropouts. In: Proceedings of the 2008 IFAC World Congress, pp. 12442–12447 (2008)
30. Ahn, H.S., Moore, K.L., Chen, Y.Q.: Stability of discrete-time iterative learning control with random data dropouts and delayed controlled signals in networked control systems. In: Proceedings the IEEE International Conference Control Automation, Robotics, and Vision, pp. 757–762 (2008)
31. Bu, X., Hou, Z.-S., Yu, F.: Stability of first and high order iterative learning control with data dropouts. Int. J. Control Autom. Syst. **9**(5), 843–849 (2011)
32. Bu, X., Yu, F., Hou, Z.-S., Wang, F.: Iterative learning control for a class of nonlinear systems with random packet losses. Nonlinear Anal. Real World Appl. **14**(1), 567–580 (2013)
33. Bu, X., Hou, Z.-S., Yu, F., Wang, F.: H-∞ iterative learning controller design for a class of discrete-time systems with data dropouts. Int. J. Syst. Sci. **45**(9), 1902–1912 (2014)
34. Bu, X., Hou, Z.-S., Jin, S., Chi, R.: An iterative learning control design approach for networked control systems with data dropouts. Int. J. Robust Nonlinear Control **26**, 91–109 (2016)
35. Huang, L.-X., Fang, Y.: Convergence analysis of wireless remote iterative learning control systems with dropout compensation. Math. Probl. Eng. **2013**, 609284 (2013)
36. Liu, J., Ruan, X.: Networked iterative learning control approach for nonlinear systems with random communication delay. Int. J. Syst. Sci. **47**(16), 3960–3969 (2016)
37. Liu, C., Xu, J.-X., Wu, J.: Iterative learning control for remote control systems with communication delay and data dropout. Math. Probl. Eng. **2012**, 705474 (2012)
38. Shen, D., Wang, Y.: ILC for networked discrete systems with random data dropouts: a switched system approach. In: Proceedings of the 33rd Chinese Control Conference, pp. 8670–8677 (2014)
39. Shen, D., Zhang, C., Xu, Y.: Two compensation schemes of iterative learning control for networked control systems with random data dropouts. Inf. Sci. **381**, 352–370 (2017)
40. Shen, D., Zhang, C., Xu, Y.: Intermittent and successive ILC for stochastic nonlinear systems with random data dropouts. Asian J. Control (2018). https://doi.org/10.1002/asjc.1480
41. Shen, D., Xu, J.-X.: A novel Markov chain based ILC analysis for linear stochastic systems under general data dropouts environments. IEEE Trans. Autom. Control **62**(11), 5850–5857 (2017)
42. Pan, Y.-J., Marquez, H.J., Chen, T., Sheng, L.: Effects of network communications on a class of learning controlled non-linear systems. Int. J. Syst. Sci. **40**(7), 757–767 (2009)
43. Shen, D., Wang, Y.: Iterative learning control for networked stochastic systems with random packet losses. Int. J. Control **88**(5), 959–968 (2015)

44. Shen, D., Wang, Y.: ILC for networked nonlinear systems with unknown control direction through random Lossy channel. Syst. Control Lett. **77**, 30–39 (2015)
45. Saab, S.S.: A discrete-time stochastic learning control algorithm. IEEE Trans. Autom. Control **46**(6), 877–887 (2001)
46. Hassibi, A., Boyd, S.P., How, J.P.: Control of asynchronous dynamical systems with rate constraints on events. In: Proceedings the 38th IEEE Conference on Decision and Control, pp. 1345–1351 (1999)
47. Shen, D., Chen, H.-F.: Iterative learning control for large scale nonlinear systems with observation noise. Automatica **48**, 577–582 (2012)
48. Li, X.D., Chow, T.W.S., Ho, J.K.L.: 2D system theory based iterative learning control for linear continuous systems with time delays. IEEE Trans. Circuits Syst. **52**(7), 1421–1430 (2005)
49. Meng, D., Jia, Y., Du, J., Yu, F.: Robust iterative learning control design for uncertain time-delay systems based on a performance index. IET Control Theory Appl. **4**(5), 759–772 (2010)
50. Shen, D., Mu, Y., Xiong, G.: Iterative learning control for non-linear systems with deadzone input and time delay in presence of measurement noise. IET Control Theory Appl. **5**(12), 1418–1425 (2011)
51. Seel, T., Schauer, T., Raisch, J.: Iterative learning control for variable pass length systems. In: Proceedings of the 18th IFAC world congress, pp. 4880–4885 (2011)
52. Seel, T., Werner, C., Schauer, T.: The adaptive drop foot stimulator - multivariable learning control of foot pitch and roll motion in paretic gait. Med. Eng. Phys. **38**(11), 1205–1213 (2016)
53. Seel, T., Werner, C., Raisch, J., Schauer, T.: Iterative learning control of a drop foot neuroprosthesis - generating physiological foot motion in paretic gait by automatic feedback control. Control Eng. Pract. **48**, 87–97 (2016)
54. Seel, T., Schauer, T., Raisch, J.: Monotonic convergence of iterative learning control systems with variable pass length. Int. J. Control **90**(3), 393–406 (2017)
55. Li, X., Xu, J.-X., Huang, D.: An iterative learning control approach for linear systems with randomly varying trial lengths. IEEE Trans. Autom. Control **59**(7), 1954–1960 (2014)
56. Li, X., Xu, J.-X., Huang, D.: Iterative learning control for nonlinear dynamic systems with randomly varying trial lengths. Int. J. Adap. Control Signal Process. **29**(11), 1341–1353 (2015)
57. Li, X., Xu, J.-X.: Lifted system framework for learning control with different trial lengths. Int. J. Autom. Comput. **12**(3), 273–280 (2015)
58. Shen, D., Zhang, W., Wang, Y., Chien, C.-J.: On almost sure and mean square convergence of p-type ILC under randomly varying iteration lengths. Automatica **63**, 359–365 (2016)
59. Shen, D., Zhang, W., Xu, J.-X.: Iterative learning control for discrete nonlinear systems with randomly iteration varying lengths. Syst. Control Lett. **96**, 81–87 (2016)
60. Li, X., Shen, D.: Two novel iterative learning control schemes for systems with randomly varying trial lengths. Syst. Control Lett. **107**, 9–16 (2017)
61. Shi, J., He, X., Zhou, D.: Iterative learning control for nonlinear stochastic systems with variable pass length. J. the Frankl. Inst. **353**, 4016–4038 (2016)
62. Wei, Y.-S., Li, X.-D.: Varying trail lengths-based iterative learning control for linear discrete-time systems with vector relative degree. Int. J. Syst. Sci. **48**(10), 2146–2156 (2017)
63. Liu, S., Debbouche, A., Wang, J.: On the iterative learning control for stochastic impulsive differential equations with randomly varying trial lengths. J. Comput. Appl. Math. **312**, 47–57 (2017)
64. Liu, S., Wang, J.: Fractional order iterative learning control with randomly varying trial lengths. J. the Frankl. Inst. **354**, 967–992 (2017)