

# 第 1 章 随机迭代学习控制

本章综述了随机迭代学习控制方面的相关研究进展. 迭代学习控制适用于可以不断重复完成指定任务的系统. 在过去三十年中, 迭代学习控制在理论和实际两方面都取得了许多重要的进展. 而在迭代学习控制中, 与随机信号有关的研究成果还比较少. 这里的随机信号包括系统噪声、量测噪声、随机数据丢包等各种在实际系统中普遍存在的信号. 本章从关键技巧的角度综述了相关进展, 包括三个方面: 线性随机系统、非线性随机系统和其他随机信号. 进而介绍了几个有潜力的研究方向, 包括点对点迭代学习控制、变轨迹迭代学习控制以及分散式/分布式迭代学习控制.

## 1.1 迭代学习控制

在生活中, 有这样一种认识, 一件事情重复去做, 通常会做得越来越好. 例如定点投篮, 当一个人进行定点投篮时, 一开始他可能投不中, 但随着不断练习, 他很有可能很快就会命中. 这里一个很重要的原因就是, 投篮者可以不断地从已有经验中学习. 具体地说, 投篮者可以根据之前投篮的偏差来调整下一次投篮的角度和力度, 从而使得篮球逐渐接近篮筐乃至命中. 迭代学习控制便是基于这样一种朴素的想法发展起来的控制分支. 粗略地说, 迭代学习控制是通过对之前过程信息的学习来改善系统的表现性能的.

因为要不断重复学习, 迭代学习控制适用于运行过程能够不断重复的系统, 通常用于跟踪某一给定目标且跟踪目标往往保持不变. 对这样的系统, 迭代学习控制能够根据之前的输入和输出数据, 结合跟踪目标来构造应用到下一次运行过程的输入信号, 从而达到逐步改善系统跟踪性能的目标. 由此可知, 迭代学习控制具有如下几个特点: ① 系统能够在有限的时间长度运行完毕并不断重复; ② 系统能够重复回到同一初始值; ③ 系统重复跟踪同一目标轨迹等. 迭代学习控制的基本思想如图 1.1 所示.

在图 1.1 中, 对跟踪目标  $y_d$ , 在第  $k+1$  次运行过程中, 根据第  $k$  次过程的输入  $u_k$  及跟踪情况  $e_k = y_d - y_k$  构造新的输入  $u_{k+1}$ . 这个输入  $u_{k+1}$  即应用到第  $k+1$  次运行过程的输入, 同时  $u_{k+1}$  也存储到记忆器中用于下一次运行过程输入的构造, 系统在输入  $u_{k+1}$  作用下输出信号为  $y_{k+1}$ , 其与跟踪目标  $y_d$  的差值也被用于构造下一次过程的输入. 至此, 系统沿迭代过程形成闭环反馈.

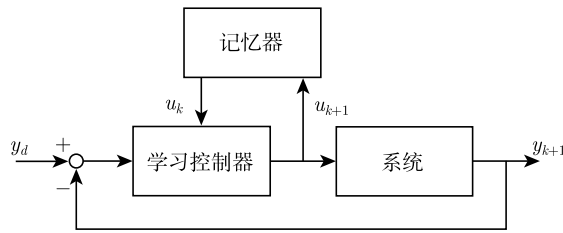


图 1.1 迭代学习控制示意图

将迭代学习控制与现实生活类比来看,系统当前及以前的输入/输出信息相当于现实生活中的经验.在现实生活中,我们就是根据这些经验制定出下一次执行相同任务时的策略,而这种新的策略就相当于在迭代学习控制中应用到下一运行过程中的输入信号.在现实中,以往的经验会改善我们做事的表现,那么在系统控制之前的运行情况一定程度上也会帮助改善我们对系统的控制性能.

迭代学习控制的主要优势在于其控制律的设计仅要求系统的跟踪目标及输入/输出数据.换言之,关于系统自身的信息要求很少,甚至可以完全未知.迭代学习控制算法形式简单且行之有效.

注意到迭代学习控制是一种沿着运行过程调整而非沿运行时间调整的控制方法,这恰是迭代学习控制不同于非学习型控制的本质所在.例如,反馈控制是一种典型的非学习型控制,迭代学习控制与反馈控制的不同之处在于,反馈控制没有利用之前运行过程的情况,仅根据当前运行过程的输入/输出数据进行调整,在每个运行过程可能产生相同的跟踪误差,而迭代学习控制则充分学习之前运行过程的跟踪情况,在其控制中包含了对已有信息的学习与利用.此外,迭代学习控制与具有一定学习能力的自适应控制也有区别,自适应控制通常是针对相对固定的控制器来不断调整其控制器参数,而迭代学习控制则是构造新的输入信号.

迭代学习控制的概念可以追溯到 Uchiyama 于 1978 年发表的文献 [1], 但该文为日文文献, 未获得广泛传播. 而在 1984 年的文献 [2]~[4] 开启了对迭代学习控制的研究. 自此以后, 大量关于迭代学习控制的文献涌现出来, 尤其是在 20 世纪末 21 世纪初研究活跃. 相关的专著有文献 [5]~[12], 综述文章包括文献 [13]~[15] 等. 部分期刊也组织了一些期刊专号, 如文献 [16]~[18]. 时至今日, 迭代学习控制已经成为一个重要的控制分支, 并在多种实际应用中取得了良好的控制效果<sup>[9]</sup>. 迭代学习控制在实际中的应用包括机器人<sup>[19-22]</sup>、硬盘驱动<sup>[23, 24]</sup>、工业/化工过程<sup>[25, 26]</sup> 等.

### 1.1.1 迭代学习控制基础

本节给出迭代学习控制的基本模型设定, 并给出一些传统的收敛结果. 考虑如

下离散时间线性时不变系统

$$\begin{aligned} x(t+1, k) &= Ax(t, k) + Bu(t, k) \\ y(t, k) &= Cx(t, k) \end{aligned} \quad (1.1)$$

其中,  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^p$  和  $y \in \mathbb{R}^q$  分别表示系统的状态、输入和输出; 矩阵  $A$ 、 $B$  和  $C$  是具有适当维数的系统矩阵;  $t$  代表一个运行批次内的任意时刻,  $t = 0, 1, \dots, N$ , 这里  $N$  表示系统运行的时间长度; 而  $k$  表示不同的运行过程, 取值为  $k = 0, 1, \dots$ . 为方便, 对离散时间系统的  $t$  取值  $0 \sim N$ , 用记号  $t \in [0, N]$  表示.

由于迭代学习控制要求系统能够不断重复运行, 所以系统需要不断重置. 在基本模型及众多研究中, 需要下述初始值重置条件

$$x(0, k) = x_0, \quad \forall k \quad (1.2)$$

系统的跟踪目标为事先给定的目标轨迹  $y(t, d)$ ,  $t \in [0, N]$ . 注意到初始值重置条件, 一般要求  $y(0, d) = y_0 \triangleq Cx_0$ . 迭代学习控制的设计目标即通过不断更新系统的输入  $u(t, k)$ , 使得相应输出  $y(t, k)$  尽可能地接近于  $y(t, d)$ . 为此,  $\forall t \in [0, N]$ , 记跟踪误差为

$$e(t, k) = y(t, d) - y(t, k) \quad (1.3)$$

因此, 迭代学习控制的设计实际上是基于之前运行过程的输入  $u(t, k)$  与跟踪误差  $e(t, k)$  来构造控制更新律, 以得到下一次运行过程的系统输入信号. 这一关系在广义上可以写为

$$u(t, k+1) = h(u(\cdot, k), \dots, u(\cdot, 0), e(\cdot, k), \dots, e(\cdot, 0)) \quad (1.4)$$

当上述函数关系仅依赖于上一过程时, 称为一阶迭代学习控制律; 其余的则称为高阶迭代学习控制律. 一般而言, 要同时兼顾算法的简单性与有效性, 因此很多迭代学习控制都设计为一阶形式, 即

$$u(t, k+1) = h(u(\cdot, k), e(\cdot, k)) \quad (1.5)$$

特别地, 控制律多设计为线性形式. 其中, 最简单的迭代学习控制律为

$$u(t, k+1) = u(t, k) + Ke(t+1, k) \quad (1.6)$$

在这一更新律中,  $K$  为学习增益矩阵, 它是这一算法的设计目标;  $u(t, k)$  表示上一运行过程的输入; 而  $Ke(t+1, k)$  表示基于上一过程跟踪误差的修正项. 注意到在系统 (1.1) 中  $t$  时刻的输入直接影响到  $t+1$  时刻的状态, 进而影响到  $k+1$  时刻的

输出, 因此在算法 (1.6) 中用  $t+1$  时刻的跟踪误差来修正  $t$  时刻的输入信号. 通常, 称算法 (1.6) 为 P 型迭代学习控制律. 当修正项变为  $K(e(t+1, k) - e(t, k))$  时, 相应的控制律称为 D 型迭代学习控制律.

对系统 (1.1) 及更新算法 (1.6), 一个基本的收敛结果为: 若  $K$  满足

$$\|I - CBK\| < 1 \quad (1.7)$$

则有  $\|e(t, k)\| \xrightarrow[k \rightarrow \infty]{} 0$ . 其中,  $\|\cdot\|$  为算子范数.

从上述收敛结果可以看出,  $K$  的设计并不需要已知关于系统矩阵  $A$  的任何信息, 而仅依赖于矩阵乘积  $CB$ . 这反映了迭代学习控制律的设计优势, 它对系统自身的信息依赖性很小. 因此可以用于处理未知性较多的系统跟踪问题.

**注记 1.1** 从迭代学习控制的表示形式可以看出, 其模型具有典型的 2D 系统特征. 很多学者便从这一点入手, 发展起了一套基于 2D 系统的研究方法, 成为研究迭代学习控制的主要方法之一.

注意到系统总是运行有限时间长度  $N$  后便重置并再次重复, 因此可以把这有限个时刻的输入与输出信号通过堆积的办法构造成如下形式, 称为超向量 (super-vector)

$$U_k = [u^T(0, k), u^T(1, k), \dots, u^T(N-1, k)]^T \quad (1.8)$$

$$Y_k = [y^T(1, k), u^T(2, k), \dots, y^T(N, k)]^T \quad (1.9)$$

记

$$G = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{N-1}B & CA^{N-2}B & \dots & \dots & CB \end{bmatrix} \quad (1.10)$$

则易知

$$Y_k = GU_k + \mathbf{d} \quad (1.11)$$

其中

$$\mathbf{d} = [(Cx_0)^T, (CAx_0)^T, \dots, (CA^N x_0)^T]^T \quad (1.12)$$

类似于式 (1.8) 和式 (1.9), 可记

$$Y_d = [y^T(1, d), u^T(2, d), \dots, y^T(N, d)]^T$$

$$E_k = [e^T(1, k), e^T(2, k), \dots, e^T(N, k)]^T$$

则有

$$U_{k+1} = U_k + \mathbf{K}E_k \quad (1.13)$$

其中,  $\mathbf{K} = \text{diag}\{K, K, \dots, K\}$ . 此时

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

由此不难看出前面所述迭代学习控制收敛的充分条件, 即式 (1.7). 不仅如此, 从上式还可以看出, 保证跟踪误差渐近收敛所需条件为  $\mathbf{K}$  的谱半径小于 1, 换言之, 即  $\mathbf{K}$  的谱半径小于 1. 实际上, 堆积模型的意义并不仅仅在于帮助我们得到收敛条件, 更重要的是, 它使得我们可以更为本质地看待迭代学习控制. 在堆积模型 (1.11) 中, 同一迭代过程内的演化过程被集成到各个记号中, 从而突出了迭代过程的意义所在. 换言之, 堆积模型 (1.11) 实际上是一个沿  $k$  变化的过程, 而  $t$  此时已经没有任何影响.

**注记 1.2** 由于迭代学习控制的关注点集中在如何沿迭代过程逐步改善系统的跟踪性能. 从其控制律的设计模式或上述基于超向量的模型 (1.11) 都可以看出这一点. 因此粗略地说, 系统由线性时不变模型拓展到线性时变模型一般不会产生新的难点. 这是因为, 当固定一个时刻后, 沿迭代过程观察在这一点上的更新情况, 本质上是一个时不变模型.

跟踪目标  $y(t, d)$  通常为可实现的, 即对合适的初始值  $x_0$ , 存在输入  $u(t, d)$ , 使得式 (1.1) 中的  $k$  替换为  $d$  仍成立. 换言之,  $Y_d = GU_d + \mathbf{d}$ , 其中  $U_d$  的定义类似于式 (1.8). 那么研究输出收敛到目标轨迹的问题  $\lim_{k \rightarrow \infty} Y_k = Y_d$ , 便可转化为研究输入收敛到目标输入的问题  $\lim_{k \rightarrow \infty} U_k = U_d$ . 对输出含有随机噪声的系统, 这种方法更为方便.

**注记 1.3** 若跟踪目标不可实现, 这意味着不存在控制输入能够产生目标轨迹, 即无论如何是不能实现完全跟踪的. 此时迭代学习控制律的设计目标也不再是实现渐近精确跟踪目标, 而是渐近收敛到一个距离目标轨迹最近的轨迹上去. 因此, 跟踪问题变成一个优化问题. 另外, 从实际应用的角度来看, 跟踪目标通常都是可实现的, 这一假定并不苛刻.

上述讨论都是围绕线性系统情形给出的, 而针对非线性系统的迭代学习控制问题也引起了众多研究<sup>[27-29]</sup>. 因其自身内在非线性特性的制约, 针对非线性系统的迭代学习控制难度要明显高于线性系统. 一个明显的佐证是, 大多数研究非线性系统的文献都要求非线性函数满足全局 Lipschitz 条件 (global Lipschitz condition,

GLC), 而这一条件限制了相关研究结果的适用范围. 因此, 如何进一步放宽这一条件是非线性系统迭代学习控制研究的重要方向.

已有文献的研究主题涵盖了迭代学习控制的众多方面, 包括更新律的设计、初始重置条件的放宽、鲁棒性、最优性、瞬态性能表现、变化轨道的跟踪, 以及实际应用等. 这里不再一一展开细述, 有兴趣的读者可以通过相关综述文献了解研究脉络. 例如, 文献 [14] 中, 作者对 1998~2004 年的迭代学习控制文献从理论和应用两个方面进行了详细分类. 文献 [30] 从范数优化的角度综述了相关系列研究.

### 1.1.2 随机迭代学习控制

本节将给出本书中随机迭代学习控制的含义. 在本书中, 随机迭代学习控制指对模型中含有随机信号的系统的迭代学习控制. 这里随机信号指用概率论中的随机变量描述的信号, 如系统运行过程的噪声、输出量测噪声、随机延迟信号等. 用随机变量描述的信号同时具有不确定性与随机性, 其值往往没有一个明确的上界.

具体而言, 本书所述的随机迭代学习研究主要分为如下几类.

- (1) 含有系统噪声及/或量测噪声的线性系统.
- (2) 含有系统噪声及/或量测噪声的非线性系统.
- (3) 含有其他随机信号的系统, 如带随机丢包的网络控制系统、带随机异步信号的大规模系统等.

目前, 关于确定性系统已经有较为深入的研究, 得到了许多有意义的结果, 并已应用于实际工程. 然而, 在实际系统中总是不可避免地存在各种各样的扰动或噪声. 因此, 考虑扰动抑制的迭代学习控制是一个重要的研究方向. 不过, 这类研究目前多是假定系统中的扰动有界, 进而证明跟踪误差是被一个与扰动上界有关的函数界住. 具体而言, 记系统扰动为  $w_k$ , 假定  $\|w_k\| \leq \epsilon$ , 其中  $\epsilon$  为适当正数, 通过各种迭代学习控制器可以使跟踪误差  $e_k$  满足  $\|e_k\| \leq L(\epsilon)$ ,  $L(\epsilon) \xrightarrow{\epsilon \rightarrow 0} 0$ . 这方面的文献如文献 [31] 和文献 [32]. 也有部分文献考虑通过学习的方法来消除扰动. 然而这些方法都不能用于处理含有随机信号的系统.

因此, 要处理随机迭代学习控制问题, 需要在确定性的迭代学习控制研究基础上, 更多地考虑对随机变量的处理方法. 现在研究中已经得到了部分处理方法及收敛结果, 本章将在其余节中进行回顾, 但仍有大量的问题有待深入研究.

值得注意的是, 在随机迭代学习控制中, 由于有随机信号的存在, 相应的收敛结果表述往往有别于传统的收敛表述. 换言之, 在随机迭代学习控制中, 关于收敛性的表述往往与某种概率收敛结合在一起, 例如, 算法以概率 1 收敛到最优值, 算法依概率收敛到目标值, 或者算法均方收敛到最优值等. 此外, 若系统输出端含有随机噪声, 直观来看, 很难得到实际输出渐近收敛到 0 的收敛结果, 此时得到的最好的收敛结果应为除去输出噪声外, 实际输出与跟踪目标的差值渐近趋于 0. 换言

之,从长远来看,系统输出的跟踪误差应逐渐地仅由输出噪声导致.

## 1.2 线性随机系统

### 1.2.1 基于卡尔曼滤波的方法

考虑如下随机线性系统

$$\begin{aligned} x(t+1, k) &= A(t)x(t, k) + B(t)u(t, k) + w(t, k) \\ y(t, k) &= C(t)x(t, k) + v(t, k) \end{aligned} \quad (1.14)$$

其中,  $A(t)$ 、 $B(t)$  与  $C(t)$  为时变系统矩阵;  $w(t, k)$  与  $v(t, k)$  分别为系统噪声与量测噪声. 注意到系统 (1.14) 与系统 (1.1) 的区别之处除了系统矩阵变为时变外, 就是增加了噪声项, 因此, 传统的基于范数压缩的分析方法及条件 (1.7) 不再适用.

Saab 针对式 (1.14) 的迭代学习控制问题给出了一系列优秀的结果. 他在文献 [33] 中首次考虑了离散时间线性系统的随机迭代学习控制, 并在之后的一系列文献中不断完善相应的研究<sup>[34-36]</sup>. 文献 [33] 中, 更新律设计为

$$u(t, k+1) = u(t, k) + K(t, k)[e(t+1, k) - e(t, k)] \quad (1.15)$$

其中,  $K(t, k)$  为学习增益矩阵.

**注记 1.4** 实际上, 文献 [33] 中考虑的系统模型与式 (1.14) 略有差别, 不同之处在于系统量测方程除了噪声  $v(t, k)$  外, 还有一项与时间无关的噪声项  $v_b(k)$ . 但设计输入更新律时, 采用了如式 (1.15) 的 D 型迭代学习控制律, 噪声项  $v_b(k)$  被直接消掉, 并在后续收敛性分析中无任何体现, 故可以不考虑这一项, 而直接考虑式 (1.14) 所示模型.

为分析这一迭代学习控制算法的收敛性, 需要几个假设条件. 首先跟踪目标  $y(t, d)$  可实现, 需要此假设的原因及相关解释已经在注记 1.3 中给出. 换言之, 存在目标输入  $u(t, d)$  及初始状态  $x(0, d) = x_0$  使得

$$\begin{aligned} x(t+1, d) &= A(t)x(t, d) + B(t)u(t, d) \\ y(t, d) &= C(t)x(t, d) \end{aligned} \quad (1.16)$$

此外, 假设输入/输出耦合矩阵  $C(t+1)B(t)$  为列满秩. 为在不引起混淆的前提下保持符号简洁, 在本章其余部分简记符号  $C^+B = C(t+1)B(t)$ . 同时用前置符号  $\delta$  的方式来表示目标值与实际值的差距, 即  $\delta x(t, k) \triangleq x(t, d) - x(t, k)$ ,  $\delta u(t, k) \triangleq u(t, d) - u(t, k)$ . 以  $E$  表示期望. 然后, 关于噪声、初始状态及初始批次输入的假设可以明确如下<sup>[33]</sup>.  $w(t, k)$  与  $v(t, k)$  均假定为零均值的高斯白噪声, 且

协方差阵  $Q_t = E[w(t, k)w(t, k)^T]$  为半正定阵, 而  $R_t = E[v(t, k)v(t, k)^T]$  为正定阵. 同时假定  $w(t, k)$  与  $v(s, l)$  不相关,  $\forall t, s \in [0, N], k, l \in \mathbb{R}$ . 初始状态假定为随机变量, 满足  $\delta x(0, k)$  为零均值高斯白噪声且协方差阵  $P_{x,0} = E[\delta x(0, k)x(0, k)^T]$  为半正定阵. 此外,  $\delta x(0, k)$  与各种噪声均不相关. 初始批次输入需要满足初始输入偏差  $\delta u(t, 0)$  为零均值白噪声且其协方差阵  $E[\delta u(t, 0)\delta u(t, 0)^T] = P_{u,0}$  为对称正定矩阵.

**注记 1.5** 上述关于噪声、初始状态及初始批次输入的假设主要来源于卡尔曼滤波方法的要求. 在这里, 我们对初始批次输入假设给出进一步的评论. 文献 [33] 关于初始输入的假设实际上相当于要求初始输入需要在目标输入  $u(t, d)$  附近按照正态分布产生 (这里按正态分布产生可从重复试验的角度出发理解). 作者还指出, 一种简单的满足上述要求的情形是  $\delta u(t, 0) = 0, \forall t$ . 事实上, 在系统信息未知的情形下, 满足这一要求非常困难. 迭代学习控制的本质思想更希望能够对任意初始输入, 通过不断地迭代学习来得到足够良好的输入信号. 从另一方面而言,  $\delta u(t, 0) = 0, \forall t$  相当于说明初始输入就已经选择目标输入  $u(t, d)$ , 此时显然没有必要再继续更新, 因为从某种意义上而言 (例如, 从  $Ee(t, k) = 0$  且  $E[e(t, k)e(t, k)^T] = \min$  角度来看),  $u(t, d)$  已经是足够好的输入信号. Saab 的后续研究文献 [34]~文献 [38] 中同样存在这一限制. 这是一个值得研究的开放性问题.

针对系统 (1.14)、迭代学习控制律 (1.15) 及上述假设条件, 文献 [33] 通过随机最小化轨道误差得到了学习增益矩阵  $K(t, k)$  在最小二乘意义下的计算算法, 并进而证明了算法的均方收敛性. 为得到这一结果, 作者首先推导得到如下 2D-Roessor 模型

$$X^+ = \Phi X + \Gamma Z \quad (1.17)$$

其中

$$\begin{aligned} X^+ &= \begin{bmatrix} \delta u(t, k+1) \\ \delta x(t+1, k) \end{bmatrix}, \quad X = \begin{bmatrix} \delta u(t, k) \\ \delta x(t, k) \end{bmatrix} \\ Z &= \begin{bmatrix} w(t, k) \\ v(t+1, k) - v(t, k) \end{bmatrix} \\ \Phi &= \begin{bmatrix} I - K(t, k)C^+B & K(t, k)[C(t) - C(t+1)A(t)] \\ B(t) & A(t) \end{bmatrix} \\ \Gamma &= \begin{bmatrix} K(t, k)C(t+1) & K(t, k) \\ -I & 0 \end{bmatrix} \end{aligned}$$

进而, 对  $X^+$  的协方差阵  $P^+ \triangleq E(X^+X^{+T})$  的迹关于  $K(t, k)$  求导, 并令导数为 0,



即可得到关于  $K(t, k)$  的递推算法如下

$$K(t, k) = P_{u,k} \Xi^T (\Xi P_{u,k} \Xi^T + \Lambda_{D,k})^{-1} \quad (1.18)$$

$$P_{u,k+1} = (I - K(t, k) \Xi) P_{u,k} \quad (1.19)$$

其中,  $\Xi \triangleq C(t+1)B(t)$ ,  $\Lambda_{D,k} \triangleq (C(t) - C(t+1)A(t))P_{x,k}(C(t) - C(t+1)A(t))^T + C(t+1)Q_t C(t+1)^T + R_t + R_{t+1}$ ,  $P_{x,t} = E[\delta x(t, k)\delta x(t, k)^T]$ ,  $P_{u,k} = E[\delta u(t, k)\delta u(t, k)^T]$ .

**注记 1.6** 实际上, 在文献 [33] 中, 分析  $K(t, k)$  对  $P^+$  的影响时, 注意到这样一个事实: 学习增益矩阵  $K(t, k)$  实际上是用来更新得到  $(t, k+1)$  时的输入, 因此  $K(t, k)$  对  $(t+1, k)$  位置的状态  $x(t+1, k)$  并没有影响, 因此将  $P^+$  的迹关于  $K(t, k)$  求导相当于将  $P_{u,k+1}$  的迹关于  $K(t, k)$  求导. 事实上, 在后续研究中, 如文献 [35]~文献 [39] 中, 作者在推导  $K(t, k)$  的递推公式时, 便使用了后一种思路.

下述定理为文献 [33] 的主要收敛定理.

**定理 1.1** [33] 对系统 (1.14) 应用学习律 (1.15) 及递推算法 (式 (1.18)~式 (1.19)), 若  $C(t+1)B(t)$  列满秩, 则  $\forall k, t$ , 存在合适的范数使得  $\|I - K(t, h)C(t+1)B(t)\| < 1$ , 从而  $\|P_{u,k+1}\| < \|P_{u,k}\|$ . 进而, 当  $k \rightarrow \infty$  时, 在  $[0, N]$  上一致地有  $P_{u,k} \rightarrow 0$ ,  $K(t, h) \rightarrow 0$ .

Saab 的一系列研究文献 [34]~文献 [37] 均延续了这一研究方法: 首先建立输入差值  $\delta u(t, k)$  及状态差值  $\delta x(t, k)$  的 2D-Roessor 模型, 然后对待优化的目标协方差阵 (一般是输入误差协方差阵  $P_{u,k}$ ) 关于学习增益矩阵  $K(t, k)$  求导, 并令导数为 0, 得到关于学习增益矩阵的递推算法. 最后, 基于噪声的互不相关性及其协方差阵的 (半) 正定性证明均方收敛性.

定理 1.1 建立起了基于卡尔曼滤波的随机迭代学习控制方法框架. 由定理 1.1 可以看出, 在  $C(t+1)B(t)$  列满秩及其他合适条件下, 作者得到了关于学习增益矩阵的递推算法, 并且该增益矩阵满足类似于式 (1.7) 的压缩映射条件. 此外, 虽然学习更新律 (1.15) 为差分形式, 但定理 1.1 的结果表明算法对量测噪声并不敏感. 不过应当注意到, 为使得学习律 (1.15) 及递推算法 (式 (1.18)~式 (1.19)) 能够良好运行, 需要已知较多的系统信息, 包括系统矩阵  $A(t)$ 、 $B(t)$ 、 $C(t)$ , 噪声协方差阵  $Q_t$ 、 $R_t$ , 以及状态误差协方差阵  $P_{x,t}$  等 (见  $\Lambda_{D,k}$  的表达式). 文献 [34]~文献 [36] 在放宽其中的部分信息已知要求上作出了一些改进结果.

针对算法 (1.18) 中需要已知  $P_{x,t}$  的限制, 文献 [34] 研究了去掉这一项之后系统的收敛性. 具体而言, 将算法改写成

$$\tilde{K}(t, k) = \tilde{P}_{u,k} \tilde{\Xi}^T (\tilde{\Xi} \tilde{P}_{u,k} \tilde{\Xi}^T + \tilde{\Lambda}_D)^{-1} \quad (1.20)$$

$$\tilde{P}_{u,k+1} = (I - \tilde{K}(t, k) \tilde{\Xi}) \tilde{P}_{u,k} \quad (1.21)$$

其中,  $\tilde{\Lambda}_D = C(t+1)Q_t C(t+1)^T + R_t + R_{t+1}$ .

对由算法 (式 (1.20)~式 (1.21)) 得到的递推学习增益矩阵  $\tilde{K}(t, k)$ , 结合 D 型更新律 (1.15), 文献 [34] 证明了当  $C(t+1)B(t)$  列满秩时, 类似定理 1.1 的所有结果仍然成立.

需要特别指出, 这里  $\tilde{P}_{u,k}$  仅是为了算法递推而构造的矩阵, 并非真正的输入误差协方差阵. 记基于学习律 (1.15) 及修订算法 (式 (1.20)~式 (1.21)) 而得到的真实的误差协方差阵为  $\bar{P}_{u,k}$ . 一个有趣的结果是, 文献 [34] 证明了  $\tilde{P}_{u,k}$  与  $\bar{P}_{u,k}$  的收敛性是相互等价的, 即  $\tilde{P}_{u,k} \rightarrow 0 \Leftrightarrow \bar{P}_{u,k} \rightarrow 0$ , 且收敛速率均为迭代次数  $k$  的反比例函数.

**定理 1.2** [34] 若  $C(t+1)B(t)$  列满秩, 对系统 (1.14) 应用学习律 (1.15) 及修订算法 (式 (1.20)~式 (1.21)), 则  $\|\tilde{P}_{u,k}\| < \frac{c_1}{k}$ ,  $\|\tilde{K}_{t,k}\| < \frac{c_2}{k}$ , 其中  $c_1$  和  $c_2$  为合适常数. 不仅如此, 对真实的输入误差协方差阵  $\bar{P}_{u,k}$  同样存在合适的常数  $c_3$  使得  $\|\bar{P}_{u,k}\| < \frac{c_3}{k}$ , 从而当  $k \rightarrow \infty$  时,  $\|\bar{P}_{u,k}\| \rightarrow 0$ .

至此, 可以看出, 上述学习算法中的状态误差协方差阵的去除并不影响收敛性, 但从某种程度而言, 修正的算法得到的控制输入并不是最优的. 因此为方便表述, 将含有状态误差协方差阵的更新算法 (式 (1.18)~式 (1.19)) 称为最优学习算法, 而将不含有状态误差协方差阵的更新算法 (式 (1.20)~式 (1.21)) 称为次优学习算法. 以下在基于卡尔曼滤波的随机迭代学习控制方法中提到最优算法、次优算法时含义与此类似. 注意到在文献 [33] 和文献 [34] 中均考虑了 D 型算法, 那么对于 P 型算法是否存在类似的结果? 文献 [35] 对这一问题给出了肯定的回答.

考虑如下 P 型迭代学习控制律

$$u(t, k+1) = u(t, k) + K(t, k)e(t+1, k) \quad (1.22)$$

接下来给出关于  $K(t, k)$  的递推计算方法. 完全类似文献 [33] 和文献 [34] 中的推导过程, 可知对上述 P 型更新律, 学习增益矩阵  $K(t, k)$  的最优学习算法为

$$K(t, k) = P_{u,k} \Xi^T (\Xi P_{u,k} \Xi^T + \Lambda_{P,k})^{-1} \quad (1.23)$$

$$P_{u,k+1} = (I - K(t, k) \Xi) P_{u,k} \quad (1.24)$$

其中,  $\Lambda_{P,k} = C(t+1)A(t)P_{x,k}(C(t+1)A(t))^T + C(t+1)Q_t C(t+1)^T + R_{t+1}$ .  $K(t, k)$  的次优学习算法为

$$\tilde{K}(t, k) = \tilde{P}_{u,k} \Xi^T (\Xi \tilde{P}_{u,k} \Xi^T + \tilde{\Lambda}_P)^{-1} \quad (1.25)$$

$$\tilde{P}_{u,k+1} = (I - \tilde{K}(t, k) \Xi) \tilde{P}_{u,k} \quad (1.26)$$

其中,  $\tilde{\Lambda}_P = C(t+1)Q_tC(t+1)^T + R_{t+1}$ . 可以证明, 对上述 P 型最优学习算法及次优学习算法, 定理 1.1 与定理 1.2 的结论仍旧成立<sup>[35]</sup>.

有了 P 型算法与 D 型算法之后, 一个自然的问题是: P 型算法与 D 型算法之间是否有所联系? 事实上, 注意到算法 (式 (1.18)~式 (1.19))、算法 (式 (1.20)~式 (1.21))、算法 (式 (1.23)~式 (1.24)) 与算法 (式 (1.25)~式 (1.26)) 之间的本质差别实际在于算法中有界正定矩阵  $\Lambda$  的选取不同, 分别为  $\Lambda_{D,k}$ 、 $\tilde{\Lambda}_D$ 、 $\Lambda_{P,k}$  与  $\tilde{\Lambda}_P$ . 文献 [35] 证明了只要这一矩阵为有界对称正定, 那么在  $C(t+1)B(t)$  列满秩时, 上述算法均可保证相互等价的收敛性. 这一结果揭示了 Saab 所给出的各类学习算法之间的等价性. 换言之, 当  $C(t+1)B(t)$  时, 应用 P 型最优算法或次优算法、D 型最优算法或次优算法, 相应的输入误差协方差阵均收敛至 0, 且收敛速度为迭代次数  $k$  的反比例函数.

到目前为止, 所有的收敛结果均要求  $C(t+1)B(t)$  列满秩, 这要求系统的输入维数要不多于输出维数. 当系统的输入维数多于输出维数时, 这一列满秩条件便无法继续满足. 文献 [36] 对仅含有量测噪声的情况进行了研究. 考虑如下仅含有量测噪声的系统

$$\begin{aligned} x(t+1, k) &= A(t)x(t, k) + Bu(t, k) \\ y(t, k) &= C(t)x(t, k) + v(t, k) \end{aligned} \quad (1.27)$$

注意到式 (1.27) 与式 (1.14) 的区别之处, 一是没有系统噪声  $w(t+1, k)$ , 二是输入矩阵  $B(t) = B, \forall t \in [0, N]$ . 为得到收敛性结果, 还假定系统精确重置, 即  $\delta x(0, k) = 0, \forall k$ .

**定理 1.3**<sup>[36]</sup> 对系统 (1.27), 应用 P 型控制律 (1.22) 及最优学习算法 (式 (1.23)~式 (1.24)) 或 P 型控制律 (1.22) 及次优学习算法 (式 (1.25)~式 (1.26)), 若  $C(t+1)B$  行满秩且  $\delta x(0, k) = 0$ , 则存在合适的常数  $c_4$  使得  $\|E[z(t, k)z(t, k)^T]\| < \frac{c_4}{k}, \lim_{k \rightarrow \infty} E[z(t, k)z(t, k)^T] = 0$ , 其中  $z(t, k) = C(t+1)\delta x(t, k)$ .

**注记 1.7** 在文献 [36] 中还允许系统存在任意的相对阶. 所谓相对阶, 直观上相当于系统中一个输入影响到输出的最小“时延”. 例如, 对 SISO 系统, 若相对阶为  $\mu$ , 则  $t$  时刻的输入  $u(t)$  能够直接影响到的最近输出为  $y(t+\mu)$ , 因此在构造迭代学习控制律时, 就需要用  $t+\mu$  时刻的跟踪误差  $e(t+\mu)$  来作为修正项. 在文献 [36] 中, 系统的相对阶虽然允许取任意值, 但需要事先已知, 因此相对阶并未起到本质的影响, 从而本章也未展开.

此外, 在文献 [38] 中, 作者基于文献 [33]~文献 [35] 中的系统模型, 考虑了 P 型学习控制律中遗忘因子的选取问题. 在以往的研究中, 加入遗忘因子被认为是一种较好地消除之前运行过程影响的控制方法. 这里的优化指标仍然为最小化输入误差协方差阵, 在这一指标下, 作者发现最优的遗忘因子矩阵实际为 0. 换言之, 在

传统的迭代学习控制算法中, 对于保持轨道有界与收敛性而言, 遗忘因子矩阵并不需要. 在文献 [39] 中, 作者考虑了迭代学习控制研究中的另一个重要问题, 即高阶迭代学习控制算法. 基于与文献 [38] 相同的模型与分析技巧, 作者研究了高阶迭代学习控制算法对优化系统性能的贡献. 作者证明了从最小化输入误差协方差阵的角度来看, 一阶学习算法在所有阶次学习算法中是最优的. 这是随机迭代学习控制中关于一阶学习算法与高阶学习算法性能比较的第一个结果.

### 1.2.2 基于随机逼近的方法

在 1.2.1 节中, 基于卡尔曼滤波的方法已经得到了丰富的成果, 但所有收敛性都是均方收敛意义下的结果. Chen 在文献 [40] 中首次给出了几乎必然意义 (概率 1 意义) 下的收敛结果. 其考虑模型仍为式 (1.14), 但仅考虑  $p \leq q$  的情形, 即输入维数不多于输出维数. 控制目标为使得下述平均跟踪误差最小, 即

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n \|y(t+1, k) - y(t+1, d)\|^2 = \min, \quad \text{a.s.} \quad (1.28)$$

在文献 [40] 中, 关于噪声及初始状态的假定如下.

噪声  $w(t, k)$  与  $v(t, k)$  为相互独立的零均值随机变量, 且具有有限但未知的  $2 + \delta$  阶矩, 其中  $\delta > 0$ , 协方差阵  $Q_t$  与  $R_t$  定义如前但未知. 初始状态  $x(0, k)$  与所有噪声  $w(t+1, k)$ ,  $v(t, k)$  相互独立, 且  $Ex(0, k) = x_0$ ,  $E\|x(0, k)\|^{2+\delta} < \infty$ , 其协方差阵未知.

与文献 [33] 中的噪声及初始状态条件相比, 文献 [40] 去掉了已知噪声协方差阵的限制.

跟踪目标  $y(t, d)$  为可实现信号, 即存在输入  $u(t, d)$  及初始状态  $x(0, d) = x_0$  使得式 (1.16) 成立. 在  $C(t+1)B(t)$  列满秩条件下, 通过反解上述目标模型 (1.16) 即可得出目标输入的表达式,  $\forall i = 0, 1, \dots, N-1$

$$\begin{aligned} u(i, d) = & [(C(i+1)B(i))^T (C(i+1)B(i))]^{-1} (C(i+1)B(i))^T \\ & \times (y(i+1, d) - C(i+1)A(i)x(i, d)) \end{aligned} \quad (1.29)$$

在指标 (1.28) 下, 可以证明, 任何收敛到  $u(t, d)$  的输入序列  $\{u(t, k)\}$  均为最优, 即使指标 (1.28) 取到最小值<sup>[40]</sup>.

由于文献 [40] 中对系统矩阵  $A(t)$ 、 $B(t)$ 、 $C(t)$  均假定为未知, 且噪声协方差阵  $Q_t$ 、 $R_t$  也假定为未知, 所以在构造控制算法时就需要对修正方向作出估计, 从而得到一个收敛到  $u(t, d)$  的输入序列  $\{u(t, k)\}$ . 作者基于随机逼近算法提出了一种新型的迭代学习控制算法. 为此, 先构造独立于噪声  $w(t, k)$ 、 $v(t, k)$  的向量序列  $\{\Delta(t, k)\}$ , 其中  $\Delta(t, k) = [\Delta_1(t, k), \dots, \Delta_p(t, k)]^T$  为  $p$  维向量且其每个组成元

$\Delta_j(t, k)$  为独立同分布的随机变量, 使得  $\forall k = 1, 2, \dots, t \in [0, N - 1], j = 1, \dots, p$

$$|\Delta_j(t, k)| < m, \quad \left| \frac{1}{\Delta_j(t, k)} \right| < n, \quad E \frac{1}{\Delta_j(t, k)} = 0 \quad (1.30)$$

其中,  $m, n$  为正常数, 并记

$$\bar{\Delta}(t, k) = \left[ \frac{1}{\Delta_1(t, k)}, \dots, \frac{1}{\Delta_p(t, k)} \right]^T \quad (1.31)$$

令  $\{a_k\}$ 、 $\{c_k\}$ 、 $\{M_k\}$  为满足如下条件的实数序列

$$a_k > 0, \quad a_k \xrightarrow[k \rightarrow \infty]{} 0, \quad \sum_{k=0}^{\infty} a_k = \infty \quad (1.32)$$

$$c_k > 0, \quad c_k \xrightarrow[k \rightarrow \infty]{} 0, \quad \sum_{k=0}^{\infty} \left( \frac{a_k}{c_k} \right)^{1+\frac{\delta}{2}} < \infty \quad (1.33)$$

$$M_k > 0, \quad M_{k+1} > M_k, \quad M_k \xrightarrow[k \rightarrow \infty]{} \infty \quad (1.34)$$

其中,  $\delta$  在噪声假设条件中给出. 初始输入  $u(t, 0), t \in [0, N]$  可以任意给定, 进而迭代学习控制律分奇偶迭代过程分别给出. 对奇数次迭代过程, 控制输入为上一迭代过程中控制输入再加一个微小扰动  $c_k \Delta(t, k)$ , 即

$$u(t, 2k + 1) = u(t, 2k) + c_k \Delta(t, k) \quad (1.35)$$

而对于接下来的偶数次迭代过程, 控制律由下式定义

$$\begin{aligned} \bar{u}(t, 2(k + 1)) = & u(t, 2k) - a_k \frac{\bar{\Delta}(t, k)}{c_k} (\|e(t + 1, 2k + 1)\|^2 \\ & - \|e(t + 1, 2k)\|^2) \end{aligned} \quad (1.36)$$

$$u(t, 2(k + 1)) = \bar{u}(t, 2(k + 1)) \cdot I_{[\|\bar{u}(t, 2(k + 1))\| \leq M_{\sigma_k(t)}]} \quad (1.37)$$

$$\sigma_k(t) = \sum_{l=1}^{k-1} I_{[\|\bar{u}(t, 2(l+1))\| > M_{\sigma_l(t)}]}, \quad \sigma_0(t) = 0 \quad (1.38)$$

其中,  $I_{[\cdot]}$  为示性函数, 若下标方括号内条件成立则取值 1, 否则取值 0.

注意到算法 (式 (1.35)~式 (1.38)) 实际是带扩展截断的随机逼近算法<sup>[41]</sup>. 上述算法的思想是在奇数次迭代过程中, 对输入增加一个小的扰动, 然后在偶数次迭代过程中, 通过紧邻的前两个迭代过程的跟踪误差来估计输入更新梯度. 这里,  $a_k$  相当于每次更新的步长. 与基于卡尔曼滤波方法的不同之处在于, 基于卡尔曼滤波的方法直接计算学习增益矩阵  $K(t, k)$ , 而基于随机逼近的方法则先估计更新梯度, 然后给定更新步长以保证收敛性. 对于算法 (式 (1.35)~式 (1.38)), 有如下收敛结果.

**定理 1.4**<sup>[40]</sup> 对系统 (1.14), 应用算法 (式 (1.35)~式 (1.38)), 若  $C(t+1)B(t)$  列满秩, 则产生的输入序列  $\{u(t, k)\}$  以概率 1 收敛到目标输入  $u(t, d)$ , 从而是指标 (1.28) 意义下的最优控制输入.

文献 [40] 的重要意义在于给出了一种基于随机逼近的迭代学习控制律的构造方法, 这一方法不需要已知系统矩阵及噪声的协方差阵. 在一定意义上契合了迭代学习控制尽量不依赖系统信息的控制思想, 因而系统需要通过学习来估计更新梯度. 估计的方法为随机差分, 这种方法已得到较多研究, 见文献 [42]~文献 [44].

表 1.1 给出了基于卡尔曼滤波的方法<sup>[33]</sup> 与基于随机逼近的方法<sup>[40]</sup> 的详细比较. 从表中可以看出, 两种方法的主要不同点如下.

(1) 文献 [33] 与文献 [40] 考虑的优化指标不同. 文献 [33] 考虑的指标为输入误差的协方差阵, 将此协方差阵 (本质上是关于输入的二次函数) 关于输入求导, 从而得到学习矩阵的递推算法; 而文献 [40] 则是优化二次渐近跟踪指标 (1.28). 从概率的角度来看, 前者是对随机信号取期望后的指标, 后者是对实际随机信号的衡量指标. 根据概率论中的大数定律可知, 两个指标的联系在于一定意义上二次渐近指标的极限值, 即协方差阵.

(2) 两者都是对学习增益矩阵进行估计. 不同之处在于, 文献 [33] 直接对增益矩阵基于求导进行估计, 而文献 [40] 则是固定学习增益步长, 而对梯度方向基于随机差分进行估计.

**表 1.1 基于卡尔曼滤波的方法与基于随机逼近的方法之比较**

	基于卡尔曼滤波的方法	基于随机逼近的方法
指标	输入误差协方差阵	跟踪误差的渐近二次平均指标
系统信息	已知	未知
随机噪声	零均值高斯白噪声	独立零均值随机变量且高阶矩有限
学习增益矩阵	将指标关于输入求导 并令导数为零获得	固定步长, 通过随机差分估计下降梯度
收敛性	均方意义	几乎必然意义 (概率 1 意义)

文献[45] 同样考虑系统 (1.14) 及跟踪指标 (1.28), 构造了基于随机逼近算法的迭代学习控制律. 与文献 [40] 相比, 不同之处在于: 文献 [40] 基于随机逼近中的 Kiefer-Wolfowitz(KW) 算法, 而文献 [45] 则是基于随机逼近中的 Robbins-Monro(RM) 算法. KW 算法通过随机差分来估计梯度, 而 RM 算法中没有相应的项, 因此在系统信息未知的情形下, 需要通过其他方法来估计控制方向. 具体而言, 以 SISO 系统为例, 控制方向由如下算法进行自适应调整

$$u_{k+1}(t) = u_k(t) + a_k S(p_k(t+1)) e_k(t+1) \quad (1.39)$$

$$q_{k+1}(t+1) = q_k(t+1) + \frac{1}{k+1} (e_{k+1}^2(t+1) - q_k(t+1)) \quad (1.40)$$

$$p_{k+1}(t+1) = \max\{p_k(t+1), q_{k+1}(t+1)\} \quad (1.41)$$

其中,  $q_0(t) = 0$ ,  $a_0 = 1$ ,  $a_k = \frac{1}{k}$ ,  $k \geq 1$ ;  $S(\cdot)$  表示方向切换函数, 其值为  $+1$  或  $-1$ ; 而  $p_k(t+1)$  是此切换函数的变元. 这一方法的具体原理及细节在本书第 5 章中给予详细阐述.

### 1.2.3 其他方法

针对随机线性系统, 还有部分其他文献也给出了探索. 第一种可以看作基于统计意义下的方法, 即考虑对随机变量取期望转化为确定情形再进行控制算法设计与分析的方法, 如文献 [46]~[49]. 在文献 [46] 中, 作者首先将模型转化为超向量形式

$$Y_k = GU_k + \varepsilon_k \quad (1.42)$$

其中,  $\varepsilon_k$  为堆积之后的向量噪声;  $U_k$  和  $Y_k$  定义方式如式 (1.8) 和式 (1.9). 关于噪声的假设条件如下.

噪声  $\varepsilon_k$  沿迭代过程为白噪声, 且  $E\varepsilon_k = 0$ ,  $E[\varepsilon_k \varepsilon_k^T] = V$ ,  $E[\varepsilon_k \varepsilon_{k+i}^T] = 0$ ,  $i \neq 0$ , 其中  $V$  为正定矩阵.

迭代学习控制算法为

$$U_{k+1} = U_k + LE_k \quad (1.43)$$

其中,  $L$  为学习增益矩阵,  $E_k$  为跟踪误差. 定义  $G_e = I - GL$ , 不难得出

$$E_k = G_e E_{k-1} + \varepsilon_{k-1} - \varepsilon_k \quad (1.44)$$

在证明收敛性时, 作者直接对式 (1.44) 左右两端取期望, 于是得到当  $G_e$  的谱范数  $\rho(G_e) < 1$  时,  $E_k$  的期望收敛至 0. 进而证明了  $\text{Var}[E_k]$  收敛至某一个常矩阵. 严格来说, 这种取期望的处理方式实际上并没有反映出随机迭代学习控制的本质. 这是因为在取期望后, 所处理的系统实际是确定性的模型. 从另一个角度而言, 期望收敛至 0 并不一定是一个好的收敛结果, 因为期望为 0 而方差很大的随机变量仍然可能产生很大的跟踪误差. 换言之, 跟踪效果仍然可能很差.

这个方法也被文献 [47] 使用. 在该文中, 作者考虑了由单输入单输出系统所构造得到的形如式 (1.42) 的系统模型, 其中噪声建模为零均值的弱平稳过程. 基于一般化的迭代学习控制律, 该文给出了跟踪误差的数学期望与方差表达式, 进而深入分析了含遗忘因子的算法、含衰减学习增益的算法, 以及含滤波器的算法等三种算法的跟踪性能.

文献 [48] 和文献 [49] 考虑了扰动抑制问题, 其中系统的随机噪声假定为高斯平稳过程. 在文献 [49] 中, 作者证明了迭代不变的滤波器可以渐近地使得受控信号零误差跟踪. 在文献 [48] 中, 作者给出了受控信号误差的协方差阵的表达式.

针对随机线性系统的频域方法由文献 [50] 首先引入. 该文系统考虑了含有平稳噪声的单输入单输出线性时不变系统. 基于闭环 2D 系统及频域分析方法, 作者给出了收敛速率与收敛误差频谱之间的折中点. 有色噪声的情况在该文的仿真算例中进行了尝试.

最后需要指出, 虽然到目前为止还没有相关的文章发表, 但我们认为随机自适应控制方法是一种处理随机迭代学习控制问题的有效方法. 实际上, 随机自适应控制问题已经得到了深入的研究<sup>[51]</sup>. 这里我们所指的方法是, 首先对系统的参数沿迭代轴方向进行辨识, 并基于辨识的参数给出控制信号. 这一方法或可产生随机迭代学习控制的重要成果. 考虑到自适应控制的着眼点为时间轴, 而迭代学习控制的着眼点为迭代轴, 因此两者之间仍旧存在一个缺口需要填补以得到深刻的结果.

### 1.3 非线性随机系统

相较于针对线性系统随机迭代学习控制的多方面研究, 针对非线性的相关研究进展要慢一些, 研究成果也较少. 一个可能的原因是, 随机噪声与非线性函数两个因素耦合在一起时, 还缺乏强有力的处理工具. 如果再考虑离散时间系统, 相应的处理难度又会增加. 在这一方面, 还需要进一步深入的研究.

#### 1.3.1 基于卡尔曼滤波的方法

作为一类特殊的非线性系统, 仿射非线性系统通常是我们研究非线性系统的切入点. 它具备了非线性的特点, 同时输出关于输入又呈现线性关系, 在一定程度上可以降低处理的难度. Saab 在文献 [37] 中考虑如下仿射非线性离散时间系统

$$\begin{aligned} x(t+1, k) &= f(x(t, k)) + B(x(t, k))u(t, k) \\ y(t, k) &= C(t)x(t, k) + v(t, k) \end{aligned} \quad (1.45)$$

其中,  $f(\cdot)$  为定义域为  $\mathbb{R}^n$  的向量函数. 可以看出, 这里的系统方程部分是时不变模型, 也即  $f(\cdot)$ 、 $B(\cdot)$  均与时间无关. 文献 [37] 中允许系统具有任意相对阶, 但要求相对阶已知并且假定相应输出关于输入为线性关系, 因此相对阶并未引入本质的非线性影响. 为符号简便及方便说明问题本质, 本节仅考虑相对阶为 1 的情形.

基本假设条件如下: 秩方面, 假定输入/输出耦合矩阵  $G(x) = C(t+1)B(x)$  为列满秩或行满秩. 跟踪目标方面, 假定  $y(t, d)$  可实现, 即对合适的初始值  $x(0, d)$ , 存在输入  $u(t, d)$  对无噪声的系统模型能够产生输出为  $y(t, d)$ . 函数非线性方面,  $f(\cdot)$



与  $B(\cdot)$  可以达到任意多项式增长速度, 但同时要求算子  $B(\cdot)$  与  $G(\cdot)$  有界. 噪声、初始状态及初始输入等条件类似于文献 [33].

**注记 1.8** 非线性函数  $f(\cdot)$  与  $B(\cdot)$  可以允许多项式的增长速度, 这一条件放宽了对非线性函数的要求. 在此之前的多数研究中, 均需要非线性函数满足全局 Lipschitz 条件, 而对于多项式增长速度的非线性函数而言, 全局 Lipschitz 条件显然不再成立.

在文献 [37] 中,  $y(t, k)$  为量测输出, 而实际用来跟踪目标  $y(t, d)$  的输出信号为  $C(t)x(t, k)$ . 因此在文献 [37] 中,  $e(t, k) = y(t, d) - y(t, k)$  被称为输出量测误差, 而输出跟踪误差记为  $\delta\psi(t, k) \triangleq C(t)[x(t, d) - x(t, k)]$ , 因此控制目标为产生输入序列, 使得输出跟踪误差收敛至零, 且所有轨道有界.

针对仿射非线性系统, 迭代学习控制律仍为 P 型, 即式 (1.22). 通过类似于文献 [35] 和文献 [36] 但更为细致的推导可知, 形如式 (1.25) 和式 (1.26) 的学习增益矩阵  $K(t, k)$  计算算法, 可以实现输入误差协方差阵渐近收敛到 0, 且收敛速度是迭代次数为  $k$  的反比例函数.

然而, 文献 [37] 的主要贡献不在于给出相应的算法并证明其收敛性, 而在于给出了一套算法设计与分析的框架. 具体而言, 针对 P 型迭代学习控制律 (1.22) 先给出保证所有轨道有界的充要条件, 进而给出输出跟踪误差收敛至 0 的充要条件. 这些充要条件就为如何选择合适的学习增益矩阵  $K(t, k)$  提供了指导原则. 为方便说明, 这里仅考虑  $p \leq q$  的情形, 对于  $p > q$  的情形可参见文献 [37].

记  $G_k \triangleq G(x(t, k))$ ,  $K_k \triangleq K(t, k)$ ,  $\Phi_k \triangleq I - K_k G_k$ , 那么对轨道有界, 有如下结果.

**定理 1.5** 对系统 (1.45) 及更新律 (1.22),  $\forall k, t$ , 轨道有界当且仅当存在  $c_\Sigma > 0$  使得

$$\left\| \sum_{i=0}^{k-1} \left[ \prod_{j=1}^{k-1-i} \Phi_{k-j} \right] K_i K_i^T \left[ \prod_{j=1}^{k-1-i} \Phi_{k-j} \right]^T \right\| \leq c_\Sigma \quad (1.46)$$

基于这个结果, 为使得产生的轨道有界, 设计学习增益矩阵就需要满足存在常数  $c_5 > 0$  及  $c_6 > 0$ , 使得  $\left\| \prod_{i=0}^{k-1} \Phi_{k-i-1} \right\| \leq c_5$  且  $\|K_k\| \leq \frac{c_6}{k}$ .

**注记 1.9** 需要特别指出, 文献 [37] 中对有界性的理解是基于期望的. 具体而言,  $x(t, k)$  有界指  $E[\delta x(t, k)\delta x(t, k)^T]$  有界. 严格而言, 协方差阵有界只能反映随机变量的二阶矩有限, 并不代表该随机变量有界.

对于输出跟踪, 有如下充要条件.

**定理 1.6** 对系统 (1.45) 及更新律 (1.22), 若  $\lim_{k \rightarrow \infty} \delta x(0, k) = 0$ , 则随着迭代批

次  $k$  增至无穷, 系统的状态误差、输入误差及输出跟踪误差都在均方意义下收敛至零, 当且仅当

$$\lim_{k \rightarrow \infty} K_k = 0, \quad \lim_{k \rightarrow \infty} \left[ \prod_{i=0}^{k-1} \Phi_{k-i-1} \right] = 0$$

$$\lim_{k \rightarrow \infty} \sum_{i=0}^{k-1} \left[ \prod_{j=1}^{k-1-i} \Phi_{k-j} \right] K_i K_i^T \left[ \prod_{j=1}^{k-1-i} \Phi_{k-j} \right]^T = 0$$

基于这一结果可以看出, 要保证输入误差在均方意义下以反比于迭代批次  $k$  的速率收敛至零, 学习增益矩阵的设计需要满足  $\exists c_7 > 0$  及  $c_8 > 0$ , 使得  $\|K_k\| \leq \frac{c_7}{k}$  且  $\left\| \prod_{j=1}^{k-1-i} \Phi_{k-j} \right\| \leq \frac{c_8}{k}$ . 进而要使得状态误差及跟踪误差收敛, 还需要保证  $\|\delta x(0, k)\|^2 \leq \frac{c_9}{k}$ ,  $c_9 > 0$ .

最后需要指出, 由于增加了非线性函数  $f(\cdot)$  及  $B(\cdot)$  的影响, 作者在模型 (1.45) 的状态方程中去掉了噪声项  $w(t, k)$ , 仅保留了输出方程中的量测噪声  $v(t, k)$ . 因此, 从这一角度出发还有很多工作有待完成. 而为了保证输出跟踪误差渐近趋于 0, 还需要初始状态误差也要满足衰减至零的条件, 换言之, 初始状态随着迭代次数的增加能够逐渐实现精确重置. 从某个角度而言, 由于初始状态不受输入影响, 若其不渐近精确, 则无法实现输出的渐近精确跟踪. 在含有噪声的情形下, 初始状态对跟踪效果的影响如何? 可否通过学习策略使其渐近精确重置? 目前还是未解决的问题.

### 1.3.2 基于随机逼近的方法

文献 [52] 同样考虑了仿射非线性随机系统的迭代学习控制问题, 基于随机逼近算法构造了学习律并给出了收敛性分析. 具体而言, 其考虑系统如下

$$\begin{aligned} x(t+1, k) &= f(t, x(t, k)) + B(t, x(t, k))u(t, k) + w(t+1, k) \\ y(t, k) &= C(t)x(t, k) + v(t, k) \end{aligned} \quad (1.47)$$

其中,  $f(t, x)$  和  $B(t, x)$  为时变非线性函数. 不过, 如注记 1.2 所提到的, 时变性并不起主要影响. 对这一模型, 控制目标为使指标 (1.28) 最小.

基本假定为: 函数  $f(t, x)$  与  $B(t, x)$  关于  $x$  连续, 且被一个多项式函数界住, 换言之, 存在  $c_{10}$ 、 $c_{11}$ 、 $l$  使得  $\|f(t, x)\| + \|B(t, x)\| \leq c_{10}\|x\|^l + c_{11}$ . 输入维数不多于输出维数, 即  $p \leq q$ , 且  $C(t+1)B(t, x)$  列满秩,  $\forall x \in \mathbb{R}^n$ . 噪声条件类似于文献 [40] 中的假定, 但对系统噪声  $w(t, k)$  及初始状态  $x(0, k)$  还进一步要求任意阶矩有限, 即  $E\|w(t, k)\|^r < \infty$ ,  $E\|x(0, k)\|^r < \infty$ ,  $\forall r \in \mathbb{Z}^+$ . 所有随机变量均假定为沿迭代轴  $k$  为独立同分布.

记  $P(t, x) \triangleq B^T(t, x)C^T(t+1)C(t+1)B(t, x)$ , 则  $P(t, x)$  正定. 对跟踪目标  $y(t, d)$ , 我们首先归纳地给出了使指标 (1.28) 最小的最优输入  $u^0(t)$  的形式. 令  $x^0(0, k) \equiv x(0, k)$ , 沿时间轴  $t$  可依次交叉地定义  $u^0(t)$  与  $x^0(t, k)$  如下

$$u^0(t) = -[EP(t, x^0(t, k))]^{-1} \times \{E[B^T(t, x^0(t, k))C^T(t+1)f(t, x^0(t, k))] - E[B^T(t, x^0(t, k))]C^T(t+1)y(t+1, d)\} \quad (1.48)$$

$$x^0(t+1, k) = f(t, x^0(t, k)) + B(t, x^0(t, k))u^0(t) + w(t+1, k) \quad (1.49)$$

文献 [52] 首先证明了上面定义的输入  $u^0(t)$  使得指标 (1.28) 达到最小值, 进而若输入序列  $\{u(t, k)\}$  满足  $u^0(t) - u(t, k) \xrightarrow[k \rightarrow \infty]{} 0, \forall t$ , 则  $\{u(t, k)\}$  也为最优输入序列, 即使得指标 (1.28) 取到最小值. 注意到这里并没有假设跟踪目标  $y(t, d)$  可实现, 而是先根据给出的跟踪目标  $y(t, d)$  构造出一个最优输入  $u^0(t)$ . 迭代学习控制算法仍用式 (1.35)~式 (1.38), 其中参数由式 (1.30)~式 (1.34) 给出. 可以证明由此算法得到的输入序列随着迭代次数的增加, 以概率 1 收敛到最优输入  $u^0(t)$ , 从而为最优输入序列.

注意到文献 [37] 的研究重点在于保证算法均方收敛, 学习增益矩阵应满足的充要条件; 而文献 [52] 的重点在于构造一个迭代学习控制律, 使得产生的输入序列能以概率 1 收敛到最优输入. 但无论是前者还是后者, 在系统模型中, 都蕴涵了输出关于输入为线性的要求, 这促使我们进一步考虑输出关于输入本质上为非线性关系时的情况. 文献 [53] 研究了一类符合此要求的非线性单输入单输出系统, 如图 1.2 所示. 这里, 输入信号并不直接进入系统, 而是先经过静态非线性函数, 如死区、饱和、预载等, 再进入系统. 具体表达式为

$$\begin{aligned} x(t+1, k) &= f(t, x(t, k)) + b(t, x(t, k))\eta(t, k) \\ \eta(t, k) &= \mathcal{N}(u(t, k)) \\ y(t, k) &= c(t)x(t, k) + v(t, k) \end{aligned} \quad (1.50)$$

其中,  $\eta(t, k)$  为未知的中间信号, 表示输入信号经过非线性函数后的信号;  $\mathcal{N}$  表示非线性环节, 包括死区、预载及饱和等. 这三种非线性环节在现实的各种工程系统中都十分常见, 它们的存在使得输出关于输入本质上为非线性关系.

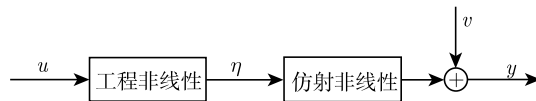


图 1.2 输入端含工程非线性的系统框图

对跟踪目标  $y(t, d)$ , 控制目标为使指标 (1.28) 达到最小值.

基本假定为: 跟踪目标  $y(t, d)$  可实现. 系统输入/输出耦合矩阵  $c(t+1)b(t, x)$  值未知, 但符号已知且不为零, 记为  $\text{sgn}(c^+b_k(t))$ . 非线性函数  $f(t, x)$  与  $b(t, x)$  关于第二个变量  $x$  被一个多项式函数界住. 量测噪声  $v(t, k)$  为零均值且二阶矩有限的随机变量, 沿迭代轴  $k$  相互独立. 初始状态渐近精确重置, 即  $\delta x(0, k) \xrightarrow[k \rightarrow \infty]{} 0$ .

注意到三种非线性环节, 即死区、预载及饱和均为非光滑非线性环节. 换言之, 这类非线性函数在某些点上不可导, 且导函数不连续, 因此文献 [40] 和文献 [52] 中通过随机差分来估计梯度方向的算法并不适合. 由于系统模型中仅有量测噪声, 构造如下算法

$$u(t, k+1) = [u(t, k) + a_k \text{sgn}(c^+b_k(t))e(t+1, k)] \times I_{[|u(t, k) + a_k \text{sgn}(c^+b_k(t))e(t+1, k)| \leq M_{\sigma_k(t)}]} \quad (1.51)$$

$$\sigma_k(t) = \sum_{i=1}^{k-1} I_{[|u(t, i) + a_i \text{sgn}(c^+b_i(t))e(t+1, i)| > M_{\sigma_i(t)}]} \quad (1.52)$$

$$\sigma_0(t) = 0 \quad (1.53)$$

其中,  $a_k$ 、 $M_k$  由式 (1.32)、式 (1.34) 给出. 上述算法实际上是带扩展截断的 RM 算法<sup>[41]</sup>. 不难看出, 上述算法中并没有涉及输入端的非线性环节  $\mathcal{N}(\cdot)$ , 换言之, 文献 [53] 针对三种非线性环节构造了统一的迭代学习控制算法. 进而证明了上述算法产生的输入序列对三种非线性环节均有界且最优.

在此研究中, 三种非光滑非线性函数的参数未知, 系统参数也未知. 然而, 算法中除了跟踪误差外, 还需要已知  $\text{sgn}(c^+b_k(t))$ , 对 SISO 系统而言, 这相当于已知系统的控制方向. 进而仅考虑量测噪声, 即在算法中噪声与非线性函数并未形成耦合, 因而可以通过随机逼近的 RM 算法递推地得到输入序列. 具体细节可参见本书第 2 章.

在上述研究基础上, Shen 与 Chen 进一步将特殊的非线性环节发展为一般的非线性环节, 并且将系统噪声也纳入考虑范围, 研究了一类 Hammerstein-Wiener 系统的迭代学习控制问题<sup>[54]</sup>. 所谓 Hammerstein-Wiener 系统是指这样一类系统, 其输入先经过一个静态非线性环节, 然后进入线性子系统, 在线性子系统的输出端又含有静态非线性环节, 如图 1.3 所示. 显然这类系统的输出关于输入为非线性关系.



图 1.3 Hammerstein-Wiener 系统模型

具体而言, 系统模型表述如下

$$\begin{aligned}
 \eta(t, k) &= f(t, u(t, k)) \\
 x(t+1, k) &= A(t)x(t, k) + B(t)\eta(t, k) + \varepsilon(t+1, k) \\
 z(t, k) &= C(t)x(t, k) + \epsilon(t, k) \\
 y(t, k) &= g(t, z(t, k)) + v(t, k)
 \end{aligned} \tag{1.54}$$

其中,  $\eta(t, k)$  与  $z(t, k)$  均为未知的中间信号; 非线性函数  $f(t, \cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^p$ ,  $g(t, \cdot) : \mathbb{R}^q \rightarrow \mathbb{R}^q, \forall t \in [0, T]$  分别表示系统输入端与输出端的静态非线性环节;  $\varepsilon(t, k)$ 、 $\epsilon(t, k)$  为系统噪声;  $v(t, k)$  为量测噪声.

跟踪目标为  $y(t, d)$ , 控制目标为使指标 (1.28) 达到最小值. 控制算法仍用带有扩展截断的随机 KW 算法 (式 (1.35)~式 (1.38)). 为保证收敛性, 需要一些基本条件, 这些基本条件可参见本书第 4 章.

需要指出, 记不存在任何噪声的情形下关于跟踪目标  $y(t, d)$  反解所得的目标输入为  $u(t, d)$ . 那么存在噪声时, 每次输入信号均选择  $u(t, d)$ , 所得输出跟踪也不能使得指标 (1.28) 达到最小. 而对于无噪声的情形,  $u(t, d)$  恰是最优输入. 这种有噪声与无噪声最优输入不同的原因在于系统噪声与非线性函数耦合, 使得系统的输出信号的期望值偏离目标值. 因此针对指标 (1.28), 需要基于期望来求取最优输入的表达式. 从这一点出发, 文献 [54] 给出了最优控制应满足的条件.

具体而言, 注意到式 (1.54) 中线性子系统部分的噪声  $\varepsilon(t, k)$ 、 $\epsilon(t, k)$  均为外加性, 故记去除噪声后的系统如下

$$\begin{aligned}
 x'(t+1, k) &= A(t)x'(t, k) + B(t)\eta(t, k) \\
 z'(t, k) &= C(t)x'(t, k)
 \end{aligned} \tag{1.55}$$

其中,  $x'(0, k) = x(0, d)$ , 则可知  $z(t, k)$  与  $z'(t, k)$  之间相差一个零均值且二阶矩有限的噪声项, 记为  $\omega(t, k)$ . 该噪声项实际上是系统噪声沿时间轴的加权叠加型, 因此关于  $k$  独立同分布. 换言之

$$y(t, k) = g(t, z'(t, k) + \omega(t, k)) + v(t, k) \tag{1.56}$$

令  $P_t(x) \triangleq E\|y(t, d) - g(t, x + \omega_t)\|^2, \forall t$ , 其中  $w_t$  为与  $\omega(t, k)$  独立同分布的随机变量. 则使得指标 (1.28) 达到最小值的中间信号  $z'(t)$  为使得  $P_t(x)$  取最小值时对应的自变量值, 进而可写出最优输入应满足的条件.

在适当的条件下, 文献 [54] 证明由算法 (式 (1.35)~式 (1.38)) 产生的输入序列以概率 1 收敛到最优控制, 从而使得渐近跟踪误差达到最小值. 具体细节可参见本书第 4 章.

### 1.3.3 其他方法

由于随机非线性系统自身所具有的难度, 相应的随机迭代学习控制成果要远少于随机线性系统的情形. 这里仅指出, 基于随机自适应控制的方法可能可以有效地解决参数化随机非线性系统的相关问题. 一种特定的情形是, 非线性系统是线性参数化的, 即系统关于未知参数为线性关系, 而其中的非线性函数则为已知的. 对于这种情形, 核心的思想为迭代更新其中的参数并进而产生相应的输入信号. 另一个具有潜力的方向为采用神经网络、模糊函数、小波函数等来逼近未知的非线性系统. 因此, 非线性系统被转化为参数线性化的情形. 本节最后将针对线性系统与非线性系统的随机迭代学习控制文献归类至表 1.2 中.

表 1.2 关于线性系统和非线性系统的随机迭代学习控制文献

	线性情形	非线性情形
KF	文献 [33]~[36]、文献 [38]、文献 [39]	文献 [37]
SA	文献 [40]、文献 [45]	文献 [52]~[54]
St	文献 [46]~[49]	
Fr	文献 [50]	

注: KF= 基于卡尔曼滤波的方法, SA= 基于随机逼近的方法, St= 基于统计意义的方法, Fr= 基于频域分析的方法

## 1.4 针对其他随机信号的迭代学习控制

### 1.4.1 随机丢包

随着网络技术的不断发展, 网络化控制系统 (networked control system, NCS) 变得日益普遍. 这类控制系统将传感器、执行器和控制器等通过网络串联起来, 增强了系统的灵活性与可靠性. 同时, 由于网络阻塞、链接中断、传输错误以及其他因素, 网络化控制系统中数据丢包是经常发生的事, 这会减弱系统的控制性能. 针对存在数据丢包的网络化控制系统的迭代学习控制问题, 目前已有部分结果. 图 1.4 为含有随机数据丢包通道的网络化控制系统框图.

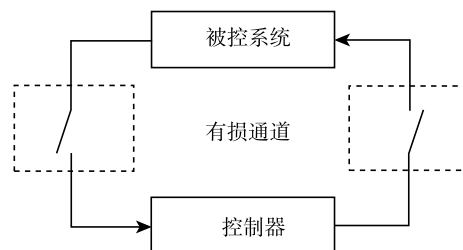


图 1.4 含丢包通道的网络化控制系统框图

这里首先给出刻画随机丢包问题的模型. 考虑系统 (1.1), 若没有随机丢包, 则更新律为式 (1.6). 现在假定数据丢包存在于输出端, 即输出信号  $y(t, k)$  在回传至控制器的过程中可能存在数据丢包. 记新的跟踪误差为  $\gamma e(t, k)$ , 其中  $\gamma \in \{0, 1\}$  是一个服从伯努利分布的二值随机变量, 用以刻画数据丢失与否. 换言之, 若数据丢包发生, 则  $\gamma = 0$ ; 若数据被成功传输至控制器, 则  $\gamma = 1$ . 令  $\bar{\gamma} = E\gamma$ .

Ahn 等针对离散时间系统, 考虑了输出端与/或输入端的随机数据丢包问题, 基于 Saab 所提出的基于卡尔曼滤波的方法给出了部分研究成果<sup>[55-57]</sup>. 因而, 所有的收敛结果都是均方收敛意义下的.

在文献 [55] 中, 系统如式 (1.1) 所示, 而更新律具有如下形式

$$u(t, k+1) = u(t, k) + K(t, k)\gamma e(t+1, k) \quad (1.57)$$

类似于文献 [33]、文献 [35]、文献 [36] 的推导步骤, 我们可以给出  $K(t, k)$  的递推计算算法. 可以证明, 只要  $\bar{\gamma} \neq 0$ , 输入误差就将均方收敛至 0. 换言之, 只要数据不是 100% 的丢包, 就可以保证类似 Saab 的结果 (如文献 [33]、文献 [35]、文献 [36]) 中的收敛性仍旧成立.

注意到在文献 [55] 中, 数据丢包是针对一个向量整体而言的, 即要么完整传输要么全部丢失. 而在实际系统中, 可能只是这个向量中的一部分数据丢失. 例如, 对  $q$  维的输出信号  $y(t, k) \in \mathbb{R}^q$ , 不妨记  $y(t, k) = [y^1(t, k), y^2(t, k), \dots, y^q(t, k)]^T$ , 可能只有其中的一部分数据发生丢失, 而另一部分数据则被传输过来. 文献 [56] 针对输入/输出维数相同的多输入多输出模型 (即  $p = q$ ), 修正输入更新律 (1.57) 为

$$u(t, k+1) = u(t, k) + K(t, k)\Gamma e(t+1, k) \quad (1.58)$$

其中,  $\Gamma$  为对角矩阵

$$\Gamma = \text{diag}\{\gamma_i\} = \begin{pmatrix} \gamma_1 & 0 & \cdots & 0 \\ 0 & \gamma_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \gamma_q \end{pmatrix} \quad (1.59)$$

其中,  $\gamma_i \in \{0, 1\}$ ,  $i = 1, \dots, p$  仍为服从伯努利分布的二值随机变量, 且相互独立. 记均值为  $\bar{\gamma}_i = E\gamma_i$ ,  $i = 1, \dots, p$ . 在这种情况下, 可以证明  $\Gamma CB$  满秩时, 输入误差协方差阵随着迭代次数的增加而趋于 0, 即输入误差在均方意义下收敛. 而  $\Gamma CB$  的满秩条件实际蕴涵了  $\Gamma$  满秩, 换言之,  $\bar{\gamma}_i \neq 0, \forall i$  时能保证均方收敛性.

无论文献 [55] 还是文献 [56], 都是假定输出数据传至控制器的过程中存在数据丢包. 在实际系统中, 从控制器将输入信号传输到系统的过程中也可能存在数据丢

包. 文献 [57] 进一步考虑了这种情形, 系统模型为不含有噪声的堆积超向量形式

$$\begin{aligned} U_{k+1} &= U_k + \Upsilon \mathcal{N}_k E_k \\ Y_k &= G \mathcal{M}_k U_k \end{aligned} \quad (1.60)$$

其中,  $\mathcal{N}_k$  与  $\mathcal{M}_k$  分别表示输出端与输入端的数据传输丢包变量, 其对角元为服从伯努利分布的二值变量, 而其余位置元为 0;  $\Upsilon$  为学习增益矩阵;  $G$  为从输入到输出的传递矩阵. 作者基于系统矩阵信息及随机矩阵  $\mathcal{N}_k$  与  $\mathcal{M}_k$  的期望矩阵, 给出了跟踪误差在均方意义下稳定的充分条件. 不过与文献 [55] 和文献 [56] 不同的是, 文献 [57] 只给出了为保证均方稳定的充分条件, 而没有给出一个可计算化的学习增益矩阵表示形式.

Bu 等从另一个角度研究了这个问题, 即通过对随机变量取期望的方法来将随机系统转化为确定系统. 文献 [58] 所考虑的系统形式为线性时不变的单输入单输出系统的超向量模型 (1.11), 其中  $d = 0$ . 相应的, 更新律形式为式 (1.57). 类似于文献 [46] 的处理技巧, 作者对迭代误差方程两端取数学期望, 可以得到期望意义下的确定型迭代误差方程, 进而可以基于这一回归方程给出一个典型的稳定性条件. 与之相对, 非线性系统情形在文献 [59] 和文献 [60] 给予了讨论, 其中系统的具体表述形式如下

$$\begin{aligned} x(t+1, k) &= f(x(t, k)) + b(x(t, k))u(t, k) \\ y(t, k) &= g(x(t, k)) + d(x(t, k))u(t, k) \end{aligned} \quad (1.61)$$

在这些非线性系统中, 所有非线性函数满足全局 Lipschitz 条件, 且跟踪目标  $y(t, d)$  假定可实现. 从而基于传统的压缩映像技巧, 在  $d(x(t, k))$  的基础上, 可以给出一个类似于文献 [58] 的稳定条件. 其原因在于, 在系统的量测方程中存在一个控制项, 而这个控制项在输出轨迹中起主导作用. 这里需要指出两点. 第一点是当系统为非线性情形时, 用以产生超向量模型的堆积技巧不再适用, 这进一步说明文献 [58] 中的技巧无法应用到文献 [59] 和文献 [60] 中的问题. 第二点是由于使用了压缩映像技巧, 作者取数学期望的对象是随机迭代不等式而不是随机迭代方程, 因而这可能会产生更为严格的稳定性条件. 简言之, Bu 系列工作的主要思想是通过数学期望来将随机过程转化为确定过程, 从而基于转化后的确定过程给出相应的稳定性条件. 进而, 文献 [61] 还针对含随机数据丢包的一类离散时间系统提供了一个  $H_\infty$  的迭代学习控制器. 借助于超向量模型, 原系统可被转化为沿迭代轴的一类离散时间随机系统, 从而可以在迭代域内定义和讨论  $H_\infty$  性能问题.

文献 [62] 针对无噪声系统也考虑了存在传输数据丢包的问题, 并且同时考虑了通信延迟的问题. 在该文中, 系统被建模为式 (1.1) 并且系统方程中含有扰动项.



数据丢包用服从伯努利分布的二值随机变量描述, 完全类似于文献 [55], 而通信延迟建模如下

$$y_c(t, k) = \pi y(t, k) + (1 - \pi)y(t - 1, k) \quad (1.62)$$

其中,  $\pi$  是取值于  $\{0, 1\}$  的服从伯努利分布的随机变量. 其中通信延迟将在后面讨论. 需要注意到数据丢包与通信延迟之间并不相互独立, 因为通信延迟只能发生在数据没有丢包时. 不过容易证明  $E[\gamma\pi] = \gamma\pi$ . 对于多维情形, 相应的数据丢包变量与通信延迟变量分别记为  $\Gamma = \text{diag}\{\gamma_i\}$ ,  $\Pi = \text{diag}\{\pi_i\}$ .

为处理数据丢包与通信延迟, 文献[62]采用了沿迭代过程进行平均的技巧, 其想法是借助历史数据来减小干扰. 为此定义迭代平均算子  $A$ : 对序列  $f_0(\cdot), f_1(\cdot), \dots, f_i(\cdot)$

$$A\{f_i(\cdot)\} = \frac{1}{i+1} \sum_{j=0}^i f_j(\cdot) \quad (1.63)$$

设计迭代学习更新律为如下含有平均算子的形式

$$u(t, k+1) = A\{u(t, k)\} + \frac{k+2}{k+1} K \Gamma \sum_{j=0}^k \tilde{e}(t+1, j) \quad (1.64)$$

其中,  $\tilde{e}(t+1, j) = y(t+1, d) - y_c(t+1, j) = \Pi e(t+1, j) + (I - \Pi)e(t, j) + (I - \Pi)\delta(t)$ ,  $\delta(t) = y(t+1, d) - y(t, d)$ ;  $K$  为学习增益矩阵. 文献 [62] 基于传统的压缩映像技巧证明了: 当学习增益矩阵满足  $\|I - K\Gamma\Pi CB\| \leq \rho < 1$  时, 学习算法 (1.64) 使得平均化后的跟踪误差期望  $E[A\{e(t, k)\}]$  收敛到一个有界范围内. 相比较而言, 这一收敛结果还不是足够优秀的, 不过借助平均化历史信号的处理技巧有望得到更好的性能表现, 这个问题还有待进一步深入研究.

针对数据随机丢包问题, 在 NCS 领域已经有很多研究, 然而基于 ILC 的数据随机丢包问题, 目前的研究才刚刚起步. ILC 与 NCS 相比, 有很多独特的优势及特点, 例如, 在 NCS 中为保证系统稳定, 数据丢包率存在一个临界值<sup>[63]</sup>, 但在 ILC 中可能并不存在这样一个值<sup>[55]</sup>. 因此, 这一方面还有许多有趣的问题等待研究.

### 1.4.2 随机异步

现实中很多工业系统往往是由多个相互关联的子系统构成的大系统. 对这类系统, 在控制子系统时往往很难获知整个大系统的所有信息, 因此常常需要构造分布式的控制器, 以较低的成本来实现较好的控制效果. 对大系统的分布式迭代学习控制, 由于各子系统的工作效率可能并不一致, 因而不能实现子系统之间输入信号的同步更新, 这就产生了研究异步式信号更新的问题. 文献 [64] 考虑了含有随机异步性的大规模系统的分布式迭代学习控制问题.

考虑由  $n$  个子系统构成的大规模系统, 其中第  $i$  个子系统表述为

$$\begin{aligned} x_i(t+1, k) &= f_i(t, x(t, k)) + b_i(t, x(t, k))u_i(t, k) \\ y_i(t, k) &= c_i(t)x_i(t, k) + v_i(t, k) \end{aligned} \quad (1.65)$$

其中,  $u_i(t, k) \in \mathbb{R}$ ,  $y_i(t, k) \in \mathbb{R}$ ,  $x_i(t, k) \in \mathbb{R}^{n_i}$ , 下标  $i$  表示子系统的编号.

每个子系统的跟踪目标分别为  $y_i(t, d)$ ,  $i = 1, \dots, n$ . 控制目标为对每个子系统, 均使得指标 (1.28) 达到最小值.

对系统的假设条件与研究思路类似于文献 [53], 但这里要考虑到各子系统更新的异步性. 这种异步性可能由子系统之间数据传输时发生丢包导致, 也可能是因为子系统之间运行效率不同导致, 往往是随机的, 无法预知的. 这里为方便表述, 在第  $k$  次更新发生时, 以  $S_k \subset \{1, \dots, n\}$  表示发生更新的那些子系统的集合, 以  $\tau(i, k)$  表示在第  $k$  次更新发生时第  $i$  个子系统已发生的更新次数之和, 即  $\tau(i, k) \triangleq \sum_{j=1}^k I_{[i \in S_j]}$ . 针对具有随机异步性的大规模系统, 第  $i$  个子系统的迭代学习控制律为

$$u_i(t, k+1) = u_i(t, k) + a(\tau(i, k))I_{[i \in S_k]}e_i(t+1, k) \quad (1.66)$$

其中,  $a(k)$  为步长, 满足

$$a(k) > 0, \quad \sum_{k=0}^{\infty} a(k) = \infty, \quad \sum_{k=0}^{\infty} a(k)^2 < \infty \quad (1.67)$$

$$a(j) = a(k)(1 + O(a(k))) \quad (1.68)$$

显然  $a(k) = \frac{1}{k+1}$  就满足上述要求.

**注记 1.10** 这里对随机异步性的描述是通过  $S_k$  和  $\tau(i, k)$  构建的.  $S_k$  是所有子系统标号集合  $\{1, \dots, n\}$  的子集, 表示在第  $k$  次更新时发生更新的子系统标号的集合. 注意到  $S_k$  是随机集合, 体现了更新的随机异步性.  $\tau(i, k)$  表示在整体系统已发生第  $k$  次更新时, 第  $i$  个子系统实际发生更新的次数, 且显然  $\tau(i, k) \leq k$ .

上述算法实际上是基于异步随机逼近算法构造的. 为保证算法能够以概率 1 收敛到最优控制, 并进而使指标 (1.28) 对每个子系统均达到最小值, 需要对异步的随机性给出如下假设.

异步性假设: 存在足够大的整数  $L$ , 使得  $\forall k, i$

$$\tau(i, k+K) - \tau(i, k) > 0 \quad (1.69)$$

通俗地说, 这一条件相当于要求任一子系统在整体系统连续发生  $L$  次更新的过程中至少要参与一次. 这实际是对子系统更新频率的一个要求.  $L$  的值并不要求已知, 只要存在即可. 如何将这一条件进一步放宽, 目前还缺少相应的研究.

### 1.4.3 随机时滞

系统中存在时滞时, 往往会对系统性能产生各种影响, 关于这一问题的研究已经是卷帙浩繁. 在 ILC 领域也有不少相关研究, 如文献 [65] 和文献 [66]. 迭代学习控制的本质是充分利用之前迭代过程中的输入与输出信息来不断调整输入信号, 以达到良好跟踪目标的任务, 那么沿迭代过程重复的信息很可能不会对迭代学习控制产生重要的影响. 因为在迭代学习过程中, 优势之一就是很少依赖于系统的具体信息, 未知的但确定的时滞从某种意义上来说可以看作未知系统的一部分. 基于这一认识, 在文献 [67] 中作者考虑了一类状态方程中含有未知时滞的仿射非线性系统, 构造了基于随机逼近算法的迭代学习控制律, 严格证明了在时滞未知的情形下, 不需要对时滞作任何估计, 仍然可以保证算法产生的输入序列以概率 1 收敛到最优控制. 广义来看, 这或许暗示系统中未知但确定的时滞在迭代学习控制方面并不像在其他时滞研究中起到本质的作用. 关于时滞对迭代学习控制的本质影响究竟如何, 目前还是一个开放性课题.

文献 [67] 的结果说明未知的但确定的时滞影响较小, 那么系统中含有随机时滞是否会产生本质的影响? 对于这一问题, 目前尚无相应的回答, 相关的研究也几乎没有. 注意到, 文献 [62] 中的通信延迟本质上就是一种随机时滞, 参见式 (1.62). 此外, 文献 [57] 中也有对随机时滞的少量研究. 在模型 (1.60) 中,  $\mathcal{N}_k$  的对角元为二值随机变量, 而其余位置的值均为 0 时, 这一矩阵度量了系统中随机数据丢包问题. 若  $\mathcal{N}_k$  表示一个广义的随机矩阵, 则其还可以表示系统的随机时滞性. 文献 [57] 实际是针对一般的随机矩阵  $\mathcal{N}_k$  给出的结论, 如前所述, 此时仅能给出均方稳定的收敛性表现.

由此可以看出, 随机时滞的出现对系统的迭代学习控制存在着本质性的影响. 这种影响有多大? 如何通过适当的控制方法消除其影响? 这些问题都还需要深入研究.

## 1.5 潜力研究方向及展望

本节将对三个有潜力的研究方向进行综述. 针对这三个研究方向, 传统的迭代学习控制研究尚未全面揭示其内在规律. 然而需要指出的是, 传统迭代学习控制中的经典问题, 如初始值重置问题、迭代变化的不确定性问题、协同迭代学习控制问题等, 都可以在随机迭代学习控制的框架下进行研究, 本节不对这些问题展开论述.

### 1.5.1 随机点对点迭代学习控制

典型的迭代学习控制要求系统输出在整个批次运行长度上完整地跟踪指定目标. 然而在很多实际问题中, 仅有部分指定位置需要实现高精度的跟踪效果, 而其

余的位置则允许有较为粗糙的跟踪效果. 例如, 一个投篮者进行定点投篮练习, 此投篮者真正关心的是其最终是否能够命中篮筐, 而并不在意篮球是否遵循某个指定的抛物线轨迹. 这类迭代学习控制问题被称为点对点迭代学习控制问题. 显然, 如果强行要求运行过程跟踪某个包含了所有指定位置的完整轨迹, 上述点对点迭代学习控制问题也会随之解决. 然而这可能会浪费控制能力, 且浪费了原本可以恰当利用的控制设计自由度.

在点对点跟踪问题中, 若仅考虑最终的位置实现高精度跟踪, 则被称为终端迭代学习控制 (terminal iterative learning control, TILC) 问题, 是一类特殊的点对点迭代学习控制问题. 在文献 [68] 中, 终端迭代学习控制被应用到晶圆制造行业中的快热化学气相沉积处理问题. 作者考虑了一类离散时间线性系统, 并且将控制输入参数化为恰当基函数的线性组合. 因此, 终端迭代学习控制针对组合系数的更新进行设计. 这一思想在文献 [69] 和文献 [70] 也有运用, 后者的不同之处在于考虑连续时间系统且基函数选择了 Legendre 正交多项式. 另一个应用研究由文献 [71] 和文献 [72] 给出, 其中终端迭代学习控制被应用到热成型及其塑料薄膜表面温度控制问题. 文献 [73] 给出了相应的高阶情形. 文献 [74] 针对一般的运行系统讨论了基于初始状态学习的终端状态控制, 提出了一种新的几何方法来证明算法的收敛性. 文献 [75] 将文献 [74] 的理论结果应用到列车到站控制, 其中使用最终停车位置误差来校正输入信号. 进而, 文献 [76] 针对线性时变多输入多输出离散系统基于输入/输出数据来构造估计和控制算法. 其中算法通过最优化二次指标函数给出. 这一得到算法的技巧同样为文献 [77] 所采用. 需要指出的是, 文献 [77] 与文献 [76] 在输入信号上有所不同, 前者在整个批次内是常值输入信号, 而后者为连续输入信号.

作为一般情形, 点对点运动控制问题已被多篇文章进行研究. 广义的点对点迭代学习控制可以根据输入与输出的位置关系划分为两类. 第一类情形是, 整个批次运行区间可以被划分为前后相连的两部分, 分别称为执行区间与观测区间. 换言之, 系统的输入信号在执行区间内起作用, 而输出轨迹的跟踪效果观测则在观测区间内完成, 彼此之间呈分离状态. 残差抑制问题即是属于此情形的一类典型点对点问题, 在文献 [78]~[80] 中有诸多研究. 在这些文章中, 使用的是传统的迭代学习控制方法, 因此未作要求的跟踪位置的自由度没有得到充分利用, 以进一步改善跟踪性能. 另一类情形是, 输入信号和输出信号是相互交错在一起的. 对于这种情形, 文献 [81] 和文献 [82] 研究了多点的点对点跟踪问题, 其控制策略为沿迭代轴方向迭代更新跟踪目标而不是输入信号, 从而有效地利用了跟踪轨迹的自由度, 而且展示了一种独特的处理点对点迭代学习控制问题的新思路. 文献 [81] 和文献 [82] 的区别之处在于, 前者使用了频域中的离散傅里叶变换技巧, 而后者使用了与前者对应的时域分析技巧. 另一种颇具潜力的处理方法为基于指定位置的跟踪信息和性能指标直接迭代更新系统的输入信号<sup>[77, 82]</sup>. 在文献 [77] 中, 作者针对多点跟踪问题

提出了一类新的迭代学习控制算法, 其中性能指标为一类仅包含指定位置跟踪误差和全部输入信号的二次指标函数, 换言之, 其指标并不是针对整个批次运行轨迹给出的. 在文献 [82] 中, 作者采取了线性参数化其控制输入信号的技巧, 其中基函数是通过系统信息矩阵进行构造的, 而性能指标则是包含了跟踪误差、控制能量和控制总量变分的广义损失函数. 此外, 文献 [83] 进一步研究了连续时间点对点跟踪问题, 给出了一种范数最优的迭代学习控制方案, 并在实验表现与理论结果之间进行了对比分析.

对多输入多输出系统而言, 上述研究都要求指定位置的输出向量信号是完整的, 即该位置的整个输出向量的每个维度都要给出其跟踪目标值, 而在实际应用中, 我们可能仅需要输出向量中的某个维度满足一定限制, 而对其他维度则可以放宽. 例如, 考虑一个三维空间中的运动问题, 此时刻画运动体位置的是一个三维向量. 在一个指定的时刻, 我们可能只会对运动体的高度有要求, 而对其水平面上的两个维度则是完全自由. 对这一类更广泛的点对点跟踪问题也有少量研究, 其中文献 [84] 针对线性系统、文献 [85] 针对非线性系统分别给出了初步尝试的结果. 文献 [84] 还深入地探索了基于梯度下降和牛顿迭代的两种迭代学习控制算法, 及其进一步向含有混合限制情形的延伸. 对这些研究, 读者还可以在文献 [86] 中阅读相关的理论结果和实验验证.

然而, 在上述文献中没有考虑随机噪声. 关于随机点对点跟踪问题的随机情形首先在文献 [87] 中给予研究. 在该文中, 作者考虑了含有系统噪声和量测噪声的线性随机系统, 其随机点对点跟踪问题的表述通过文献 [84] 的变形形式给出. 该文提出了含有衰减增益的 P 型迭代学习控制更新算法, 并且针对修正的跟踪误差给出了几乎必然收敛性的分析与证明. 可以看出, 关于随机点对点迭代学习控制的研究还处于起步阶段, 尚有许多工作有待完成.

### 1.5.2 变跟踪目标的迭代学习控制

由于任何学习策略都需要一个内在要求, 即重复性, 所以在传统迭代学习控制中, 跟踪目标在各个批次内保持不变是一个基本前提. 然而, 这可能在一定程度上限制了迭代学习控制的广泛应用, 因为当跟踪目标发生变化时, 迭代学习控制算法需要从头开始学起, 因而浪费了此前已经学习到的经验. 这一点驱动学术界进一步考虑针对变跟踪目标的学习跟踪目标, 换言之, 当跟踪目标沿迭代轴发生变化时, 学习控制算法如何充分利用已有的信息进行调整, 但是相关的研究进展到目前为止还很有限. 其原因在于对跟踪目标发生变化时, 已经获得的信息该如何利用还没有深刻的理解.

针对变跟踪目标跟踪问题, 文献 [88] 是一个早期的研究结果. 具体而言, 文献 [88] 针对连续时间非线性系统, 考虑了缓变化的跟踪目标对迭代学习控制的影响及

其解决方案. 这里, “缓变化”是指跟踪目标轨迹在前一批次与当前批次的差值存在一个小的上界. 对于这一问题, 该文提出了含有遗忘因子的更新律, 并且证明了算法的鲁棒性及有界收敛性. Xu 等对一类迭代变跟踪目标的跟踪问题给出了一种直接学习控制的方法<sup>[89-91]</sup>. 这里的变跟踪目标具体是指所有跟踪目标具有相同的形状, 但具有不同的尺度<sup>[89, 91]</sup>或具有不同的时间长度<sup>[90]</sup>. 可以看出, 相同的跟踪目标形状给直接迭代更新律提供了成功的基础. 此外, 文献 [92] 和文献 [93] 针对大尺度工业系统同样考虑了跟踪目标不重复的问题, 给出了一类分散式学习算法, 并证明了算法在  $L_p$  范数意义下的收敛性. 上述所有结果都着眼于直接型的迭代学习控制算法.

处理变跟踪目标的跟踪问题的另一条可行的途径是, 通过迭代学习系统中的不变因素, 如系统参数等, 而不是直接迭代更新控制信号. 基于这一思想同样有少量研究结果. 文献 [94] 研究了一类参数化的非线性系统, 基于等价原则给出了一个参数化的控制律, 其中参数沿批次轴进行迭代更新. 这一方法的收敛性和有效性可以通过复合能量函数 (composite energy function, CEF) 方法进行证明. Chi 等也讨论了一类参数化高阶系统的学习跟踪问题<sup>[95]</sup>, 其采用自适应迭代学习控制算法来进行参数估计及设计控制算法. 这一结果被进一步推广至系统参数为迭代变化的情形, 不过假定参数符合二阶内模原理<sup>[96]</sup>. 在该文中, 作者同样使用复合能量函数方法来给出算法的收敛性分析. 对非参数化的非线性系统, 一个直观的想法是引入全局逼近器来逼近非线性函数. 这一想法为文献 [97] 所采用, 其采用模糊系统来逼近系统的非线性, 并进而基于这一模糊系统来设计自适应迭代学习控制. 广义而言, 在运行过程中学习系统的不变因素而不是更新控制信息, 将会对变跟踪目标的跟踪问题有着重要意义.

然而, 针对变跟踪目标的跟踪问题还没有关于随机迭代学习控制方面的结果. 而在这一点上, 需要同时考虑变跟踪目标与随机噪声相耦合的情形. 针对这一开放性问题, 前述随机迭代学习控制方法, 无论是面向线性系统还是面向非线性系统的方法, 还需要作出相应的调整.

### 1.5.3 分散式/分布式协同随机迭代学习控制

多系统是指由多个子系统按照固定或者变化的拓扑关系关联在一起的大系统. 这类系统的控制和跟踪已经成为控制界的重要方向. 巡航卫星、大规模工业系统以及多智能体系统 (multi-agent system, MAS) 都是典型的多系统. 由于每个子系统往往跟踪各自的特定目标, 因而多系统往往意味着多目标跟踪问题<sup>[98-106]</sup>.

文献 [98] 和文献 [99] 研究了状态估计问题. 在文献 [98] 中, 针对多智能体系统, 作者综合比较了系统状态的联合估计和独立估计的性能, 证明联合估计仅在一些特定的条件下优于独立估计的结果. 更进一步, 联合估计的性能依赖于多智能体

系统中各个体之间的相似性. 而联合估计的敏感性问题在文献 [99] 中给出.

Meng 等在文献 [100] 中给出了多智能体系统的有限时间一致性问题. 在该文中, 每个子系统为线性模型且基于终端迭代学习控制给出了一类分布式协同算法. 作者证明了随着批次数增至无穷, 所有个体可以实现有限时间一致性. 进而, 上述结果被进一步推广至两个角度: 其一是个体模型被拓展至一类非线性系统<sup>[101]</sup>, 另一个角度是跟踪目标被拓展至广义轨迹<sup>[102]</sup>.

文献 [103] 讨论了多智能体系统的编队问题. 其中每个个体的跟踪目标并不要求相同. 该文提出了一类分布式协同算法, 并且基于传统的压缩映像技巧证明了算法的收敛性. 不仅如此, 文献 [104]~文献 [106] 同样考虑了编队控制问题. 在文献 [104] 中, 每个个体用连续时间仿射非线性系统刻画, 并且跟踪目标为个体之间的相对距离. 在文献 [106] 中, 每个个体用一个单积分模型刻画, 且跟踪目标为个体之间的绝对欧氏距离. 在文献 [105] 中, 作者进一步讨论了卫星的编队控制问题, 其中每颗卫星的跟踪目标为一个给定的轨迹.

注意到上述研究结果多数都是基于连续时间确定模型给出的, 这就说明还有很多有趣的问题尚未解决. 在传统的迭代学习控制问题中, 系统的运行长度要求在有限时间区间内完成并且不断重复, 这使得迭代学习控制应用于多智能体系统不同于传统的一致性问题. 所以, 到目前为止, 如何从理论分析和实际应用的角度定义分散式/分布式协同随机迭代学习控制, 还没有完善的回答. 此外, 子系统之间的传输还会引出更多有趣的问题. 这些都需要进一步研究.

## 1.6 本章小结

迭代学习控制是一种优秀的基于数据的控制方法, 具有广泛的工业应用背景. 这种方法可以较少地依赖于系统信息, 但仍能实现良好的控制性能, 自提出以来得到了深入的研究. 不过, 对含有随机信号的系统相关研究结果较少. 本章综述了近几年来在随机迭代学习控制方面已有的文献, 分如下三方面展开: 含系统噪声与/或量测噪声的线性系统的随机迭代学习控制、含系统噪声与/或量测噪声的非线性系统的随机迭代学习控制, 以及对含有其他随机信号的系统的迭代学习控制. 本章着重指出了相关研究的基本假设与研究方法, 并在比较相关研究的基础上给出了一些有待解决的公开问题和一些继续研究的切入点, 希望能促进控制界在这一方向上更大的突破.

针对含有系统噪声与/或量测噪声的线性系统或非线性系统的随机迭代学习控制, 目前已有的研究主要分为两类, 以 Saab 为代表的研究中得到均方收敛的结果, 但算法收敛速度较快; 而以 Chen 为代表的研究得到了概率 1 的结果, 但算法收敛速度较慢. 如何充分结合两种思路的优势, 给出新的随机迭代学习控制算法是一个

很有意义的研究点.

本章还进一步综述了针对其他随机信号的相关文献,并对一些潜在的研究方向进行了分析.在本章最后指出,关于随机迭代学习控制的应用研究到目前还十分有限,绝大多数还停留在数值仿真的层面上.换言之,这一点还有很大的研究空间.