

# 基于 2D 理论的点对点综合预测迭代学习控制

洪英东, 熊智华<sup>†</sup>, 江永亨, 叶昊

(清华大学自动化系, 北京 100084)

**摘要:** 基于二维系统综合预测迭代学习控制(2D-IPILC)方法, 结合轨迹更新策略研究点对点跟踪问题的控制算法. 该算法既能够充分利用点对点问题在非跟踪点的自由度, 也可以通过引入模型预测控制来提高时间轴的抗干扰能力. 由于轨迹更新中引入时变参数, 该 2D 模型为时变 2D 模型, 因此分析状态转移矩阵特性和系统全响应, 进而采用 2D 理论分析算法的收敛性和收敛条件, 并分析参数对控制效果的影响. 相比固定轨迹算法, 该算法的收敛速度更快, 稳定性比直接型优化算法更好. 最后通过仿真实例验证了所提出算法的效果.

**关键词:** 迭代学习控制; 2D 系统理论; 点对点跟踪; 轨迹更新; 预测控制; 时变模型

**中图分类号:** TP13

**文献标志码:** A

## Point-to-point integrated predictive iterative learning control based on two-dimensional theory

HONG Ying-dong, XIONG Zhi-hua<sup>†</sup>, JIANG Yong-heng, YE Hao

(Department of Automation, Tsinghua University, Beijing 100084, China)

**Abstract:** A two-dimensional based integrated predictive iterative learning control(2D-IPILC) method is applied to the point-to-point tracking problem with the trajectory updating strategy. The 2D-IPILC method can fully take advantages of the freedom brought by the untracked points, and can improve the capability of dealing with the disturbance in the time domain by combining the feature of model predictive control. As time-varying parameters are involved in the updating trajectory scheme, the model of the 2D-IPILC method is also time-varying. Therefore, the characteristics of time-varying transition matrix and full response are analyzed in detail. Furthermore, the convergence principle and conditions of the 2D-IPILC method are proved based on the 2D theory, and the effects of the control parameter setting are also discussed. The 2D-IPILC method can get faster convergence than the fixed trajectory methods, and is more robust than the direct optimization algorithm. Finally, these advantages are demonstrated by numerical simulation.

**Keywords:** iterative learning control; two-dimensional system theory; point-to-point tracking control; reference trajectory updating; predictive control; time-varying model

## 0 引言

迭代学习控制(ILC)是一种广泛应用于间歇过程或重复性操作过程的控制方法, 控制任务是在有限批次内输出能在一条期望轨迹上. 人们首先在工业机器人轨迹跟踪领域提出迭代学习控制算法<sup>[1-2]</sup>. 一般的轨迹跟踪问题通常是跟踪完整的期望轨迹, 但是在很多应用中, 被控系统只在一部分特定时间点上对输出有要求, 即所谓的点对点跟踪控制问题<sup>[3-4]</sup>, 常见的应用包括工业机械臂、起重机控制、自动化生产线等<sup>[5-6]</sup>.

最早的点对点跟踪问题主要集中于研究一类终

点跟踪问题, Xu 等<sup>[7]</sup>和 Hou 等<sup>[8]</sup>分别针对快速热处理和列车进站问题提出了 TILC 算法. 对于更一般的点对点跟踪控制问题, 常用的解决思路是设计一条经过特定跟踪点的轨迹, 从而把该问题转化为一般的全轨迹跟踪问题<sup>[9]</sup>, 这种方法主要关心的是跟踪轨迹的设计. 文献[10-11]提出了基于约束的输入设计方法, 然而这些方法没有充分利用点对点跟踪问题中那些非特定点的自由度, 从而降低了求解问题的效率. 与此不同, Freeman 等<sup>[12-13]</sup>提出了一种基于优化跟踪误差的方法来直接得到输入, 而不再需要设计初始轨迹, 这种方法获得了更多的自由度, 同时利用获得的

收稿日期: 2017-01-10; 修回日期: 2017-06-19.

基金项目: 国家自然科学基金项目(61473162).

责任编辑: 赵春晖.

作者简介: 洪英东(1990—), 男, 博士, 从事迭代学习控制及其应用的研究; 熊智华(1971—), 男, 副教授, 从事迭代学习控制等研究.

<sup>†</sup>通讯作者. E-mail: zhxiong@tsinghua.edu.cn

自由度来考虑更多的约束条件<sup>[14-15]</sup>,但是Freeman等人的研究没有考虑如何克服过程中的干扰.安通鉴等<sup>[16]</sup>提出了一种在时域基于轨迹更新的鲁棒学习控制方法来处理点对点跟踪控制问题,Freeman<sup>[17]</sup>也提出了一种频域内的轨迹更新算法.这类算法都是在批次内跟踪预先设计的全轨迹,并在批次间不断更新轨迹.主要的不同在于轨迹更新算法,此类算法可以与一般的迭代学习控制方法相结合,得到适用于点对点问题的方法.

针对一般的全轨迹跟踪问题,Chen等<sup>[18-19]</sup>提出了一种与预测控制相结合的综合迭代学习控制算法,该算法很好地处理了时间方向上的过程干扰.本文基于2D理论,在文献[18-19]的基础上,提出一种结合轨迹更新的综合预测迭代学习控制方法.此方法一方面考虑了点对点跟踪问题的额外自由度,另一方面在时间轴上引入模型预测控制,以解决系统干扰条件下的稳定性问题.与全轨迹跟踪问题不同,由于轨迹更新参数的引入,点对点跟踪问题的2D系统描述是时变的.与自适应ILC<sup>[20]</sup>不同的是,本文主要针对系统本身参数时不变而存在过程干扰的情况,而自适应ILC主要用于解决对象参数不确定性.本文采用2D系统理论分析系统的状态转移矩阵性质,基于此,从理论上分析算法的收敛条件,并通过仿真验证所提出算法的有效性,分析算法中参数选择的影响.

## 1 点对点迭代学习控制

### 1.1 问题描述

针对在固定时间内运行的线性时不变离散时间系统,考虑如下状态空间模型描述<sup>[8]</sup>:

$$\begin{cases} x(t+1, k) = A \cdot x(t, k) + B \cdot u(t, k), \\ y(t, k) = C \cdot x(t, k) + d(t, k). \end{cases} \quad (1)$$

其中: $t$ 和 $k$ 分别为采样时间和运行批次, $t \in [0, N]$ , $N$ 为采样点数; $x \in R^n$ 、 $u \in R$ 、 $y \in R$ 分别为状态、输入和输出; $A$ 、 $B$ 、 $C$ 分别为相应的参数矩阵; $d$ 为过程干扰和测量噪声.与通常的ILC全轨迹跟踪不同,点对点ILC的轨迹跟踪控制目标是对于某些特定的时间点集 $M = \{t_1, t_2, \dots, t_M\}$ , $M \leq N$ ,要求输出跟踪给定值 $y_{dM} = \{y_d(t_1), y_d(t_2), \dots, y_d(t_M)\}$ .将点对点ILC问题的跟踪误差记为 $e_{kM}$ ,则

$$e_{kM} = y_{dM} - y_{kM}, \quad (2)$$

其中 $y_{kM} = [y_k(t_1), y_k(t_2), \dots, y_k(t_M)]$ 为第 $k$ 批次在时间点集 $M$ 上的输出.针对这一类问题,可以首先设计一条经过特定跟踪点的初始参考跟踪轨迹,再利用非特定点的自由度在批次间更新参考跟踪轨迹.

更新轨迹的目的是让新的目标轨迹在非特定点上更接近系统上一批次的输出,而在特定跟踪点上保持不变,从而使算法可以更快地收敛,即

$$\|r_{k+1} - y_k\| \leq \|r_k - y_k\|, \quad (3)$$

其中 $y_k = [y_k(1), y_k(2), \dots, y_k(N)]$ 和 $r_k = [r_k(1), r_k(2), \dots, r_k(N)]$ 分别为第 $k$ 批次在所有时间点上的输出轨迹和目标轨迹. $r_k$ 采用如下插值更新算法得到<sup>[14]</sup>:

$$r_{k+1} = r_k + \Lambda_k(r_k - y_k), \quad (4)$$

$\Lambda_k = \text{diag}\{\lambda_k(1), \lambda_k(2), \dots, \lambda_k(N)\}$ , $\lambda_k(t)$ 为各采样时刻的轨迹更新系数,且取值为<sup>[16]</sup>

$$\begin{cases} -2 < \lambda_k(t) < 0, & t \notin M; \\ \lambda_k(t) = 0, & t \in M. \end{cases} \quad (5)$$

该轨迹更新方法可以满足式(3)的要求,由此可得跟踪误差为

$$e_k = r_k - y_k = r_{k-1} + \Lambda_k(r_{k-1} - y_{k-1}) - y_k. \quad (6)$$

由于在点对点ILC问题中只需要关心特定点集 $M$ 上的跟踪误差,可知

$$e_{kM} = y_{dM} - \Psi y_k. \quad (7)$$

其中

$$\begin{aligned} \Psi &= \{\psi_{ij}\}_{M \times N}; \\ \begin{cases} \psi_{ij} = 1, & j = t_i, t_i \in M; \\ \psi_{ij} = 0, & j \neq t_i. \end{cases} \end{aligned}$$

因为任意一条 $r_k$ 都经过所有的跟踪点设定值 $y_{dM}$ ,所以点对点ILC问题的跟踪误差可以写成

$$e_{kM} = y_{dM} - \Psi y_k = \Psi r_k - \Psi y_k = \Psi e_k. \quad (8)$$

### 1.2 系统描述

根据系统状态方程(1)和跟踪误差(6),可以把上述模型描述成在批次轴 $k$ 和时间轴 $t$ 上的2D Roesser模型<sup>[21]</sup>,如下所示:

$$\begin{aligned} \begin{bmatrix} \eta(t+1, k) \\ e(t, k+1) \end{bmatrix} &= \\ \begin{bmatrix} A & 0 \\ -CA & I + \lambda_k(t) \end{bmatrix} \begin{bmatrix} \eta(t, k) \\ e(t, k) \end{bmatrix} &+ \\ \begin{bmatrix} B \\ -CB \end{bmatrix} \Delta u(t-1, k) + \begin{bmatrix} 0 \\ -I \end{bmatrix} \Delta d(t, k). & \quad (9) \end{aligned}$$

其中

$$\begin{aligned} \eta(t, k) &= x(t-1, k+1) - x(t-1, k), \\ \Delta u(t-1, k) &= u(t-1, k+1) - u(t-1, k), \\ \Delta d(t, k) &= d(t, k+1) - d(t, k). \end{aligned}$$

由于上述2D Roesser模型中引入了轨迹更新参数 $\lambda_k(t)$ ,且 $\lambda_k(t)$ 随着时间 $t$ 变化,与一般的基于2D理论的迭代学习控制模型<sup>[18]</sup>不同,其状态转移矩阵特性也有相应的变化.

## 2 基于轨迹更新的2D-IPILC

引入2D系统对迭代学习控制算法进行分析的好处是可以在时间和批次轴两个方向上同时对系统性能进行分析.针对点对点跟踪问题,在轨迹更新算法的基础上,在批次内采用文献[18]中提出的综合迭代学习控制算法,该方法把比例型迭代学习控制算法与预测控制相结合,在批次方向上,通过比例型迭代学习控制算法根据上一个批次的误差确定输入的一部分,在时间方向上,根据预测误差确定输入的另一部分.从而既利用点对点问题在非特定点的自由度,也可以在时间方向上对过程干扰进行抑制.

### 2.1 时变状态转移矩阵

取2D-IPILC的控制律<sup>[18]</sup>为

$$\Delta u(t-1, k+1) = L \cdot e(t, k) + \Delta u_{\text{mpc}}(t-1, k+1). \quad (10)$$

其中: $L \cdot e(t, k)$ 为批次间的控制部分, $L$ 为学习率, $\Delta u_{\text{mpc}}(t-1, k+1)$ 为批次内由预测控制确定的部分.因此,2D-IPILC的目标是找到合适的 $L$ 和 $\Delta u_{\text{mpc}}(t-1, k+1)$ 使得算法收敛,即跟踪误差随批次逐渐减小.在没有引入时变参数 $\lambda_k(t)$ 的情况下,学习率 $L$ 的选取通常需要满足 $\|I - \text{CBL}\| < 1$ <sup>[22]</sup>.

将上述控制律应用于2D系统(9)中,同时为了简化考虑,设定轨迹更新参数不随着批次 $k$ 而变化,取为 $\lambda(t)$ ,因此可以得到

$$\begin{bmatrix} \eta(t+1, k) \\ e(t, k+1) \end{bmatrix} = \begin{bmatrix} A & BL \\ -CA & I - \text{CBL} + \lambda(t) \end{bmatrix} \begin{bmatrix} \eta(t, k) \\ e(t, k) \end{bmatrix} + \begin{bmatrix} B \\ -CB \end{bmatrix} \Delta u_{\text{mpc}}(t-1, k) + \begin{bmatrix} 0 \\ -I \end{bmatrix} \Delta d(t, k). \quad (11)$$

为方便起见,使用如下符号来表示:

$$\begin{aligned} \xi(t, k) &= \begin{bmatrix} \eta(t, k) \\ e(t, k) \end{bmatrix}, \quad \bar{\eta}(0, j) = \begin{bmatrix} \eta(0, j) \\ 0 \end{bmatrix}, \\ \bar{e}(i, 0) &= \begin{bmatrix} 0 \\ e(i, 0) \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \\ \bar{D} &= \begin{bmatrix} 0 \\ -CB \end{bmatrix}, \quad \bar{I} = \begin{bmatrix} 0 \\ -I \end{bmatrix}. \end{aligned}$$

定义2D模型(11)中的状态转移矩阵为

$$\Gamma = \begin{bmatrix} A & BL \\ -CA & I - \text{CBL} + \lambda(t) \end{bmatrix}, \quad (12)$$

并定义如下状态转移矩阵:

$$\begin{aligned} \Gamma^{1,0} &= \begin{bmatrix} A & BL \\ 0 & 0 \end{bmatrix}, \\ \Gamma_t^{0,1} &= \begin{bmatrix} 0 & 0 \\ -CA & I - \text{CBL} + \lambda(t) \end{bmatrix}. \end{aligned} \quad (13)$$

由2D理论可知<sup>[18]</sup>

$$\Gamma^{i,j} = \Gamma^{1,0} \cdot \Gamma^{i-1,j} + \Gamma_i^{0,1} \cdot \Gamma^{i,j-1}, \quad (i, j) > (0, 0). \quad (14)$$

与文献[18-19]相比,由于系统的状态转移矩阵 $\Gamma_i^{0,1}$ 中包含了时变参数 $\lambda(t)$ ,需要分析包含时变参数 $\lambda(t)$ 的2D系统零边界响应和零输入响应,进而得到系统的全响应.

依据2D系统理论进行分析,对于2D系统(11),系统的零输入响应为

$$\xi(i, j) = \Gamma^{i,j} \xi(0, 0). \quad (15)$$

进而可以得到系统的零边界响应为

$$\xi(i, j) = \sum_{k=0}^j \Gamma^{i,j-k} \cdot \bar{\eta}(0, k) + \sum_{k=0}^i \Gamma^{i-k,j} \bar{e}(k, 0), \quad (16)$$

其中 $\bar{\eta}(0, k)$ 、 $\bar{e}(k, 0)$ 分别表示2D系统的非零激励.因此,可以求得系统的全响应为<sup>[18]</sup>

$$\begin{aligned} \xi(t, k) &= \sum_{i=0}^t \Gamma^{t-i,k} \bar{e}(i, 0) + \sum_{(0,0) \leq (i,j) < (t,k)} (\Gamma^{t-i-2,k-j+1} \bar{B} + \Gamma^{t-i-1,k-j} \bar{D}) \Delta u_{\text{mpc}}(i, j) + \sum_{(0,0) \leq (i,j) < (t,k)} \Gamma^{t-i,k-j-1} \bar{I} \Delta d(i, j), \end{aligned} \quad (17)$$

其中状态转移矩阵 $\Gamma^{i,j}$ 可以由 $\Gamma_t^{0,1}$ 推导得到.进一步,提出2D系统中包含时变参数 $\lambda(t)$ 的状态转移矩阵特性的定理如下:

**定理1** 在时变2D系统(11)中,如果 $\forall t \in [0, N]$ ,有 $\|\Gamma_t^{0,1}\| < 1$ ,则 $\lim_{k \rightarrow \infty} \Gamma^{i,j} = 0$ .

为了证明定理1,首先证明如下两个引理.

**引理1** 对于形如式(10)的2D系统,有

$$\Gamma^{i,j} = \sum_{k=0}^j (\Gamma_i^{0,1})^k \cdot \Gamma^{1,0} \cdot \Gamma^{i-1,j-k}. \quad (18)$$

**证明**

$$\Gamma^{i,j} = \Gamma^{1,0} \cdot \Gamma^{i-1,j} + \Gamma_i^{0,1} \cdot \Gamma^{i,j-1} =$$

$$\begin{aligned} & \Gamma^{1,0} \cdot \Gamma^{i-1,j} + \\ & \Gamma_i^{0,1} \cdot (\Gamma^{1,0} \cdot \Gamma^{i-1,j-1} + \Gamma_i^{0,1} \cdot \Gamma^{i,j-2}) = \\ & \Gamma^{0,0} \cdot \Gamma^{1,0} \cdot \Gamma^{i-1,j} + \Gamma_i^{0,1} \cdot \Gamma^{1,0} \cdot \Gamma^{i-1,j-1} + \\ & \Gamma^{0,1} \cdot \Gamma_i^{0,1} \cdot (\Gamma^{1,0} \cdot \Gamma^{i-1,j-2} + \Gamma_i^{0,1} \cdot \Gamma^{i,j-3}) = \\ & \dots = \\ & \sum_{k=0}^j (\Gamma_i^{0,1})^k \cdot \Gamma^{1,0} \cdot \Gamma^{i-1,j-k}. \quad \square \end{aligned}$$

引理2 对于形如式(10)的2D系统,有

$$\Gamma^{i,j} = \sum_{h=0}^i \Gamma^{h,0} \cdot \Gamma_{i-h}^{0,1} \cdot \Gamma^{i-h,j-1}. \quad (19)$$

证明

$$\begin{aligned} & \Gamma^{i,j} = \Gamma_i^{0,1} \cdot \Gamma^{i,j-1} + \Gamma^{1,0} \cdot \Gamma^{i-1,j} = \\ & \Gamma_i^{0,1} \cdot \Gamma^{i,j-1} + \\ & \Gamma^{1,0} \cdot (\Gamma_{i-1}^{0,1} \cdot \Gamma^{i-1,j-1} + \Gamma^{1,0} \cdot \Gamma^{i-2,j}) = \\ & \Gamma^{0,0} \cdot \Gamma_i^{0,1} \cdot \Gamma^{i,j-1} + \Gamma^{1,0} \cdot \Gamma_{i-1}^{0,1} \cdot \Gamma^{i-1,j-1} + \\ & \Gamma^{1,0} \cdot \Gamma^{1,0} \cdot (\Gamma_{i-2}^{0,1} \cdot \Gamma^{i-2,j-1} + \Gamma^{1,0} \cdot \Gamma^{i-3,j}) = \\ & \dots = \\ & \sum_{h=0}^i \Gamma^{h,0} \cdot \Gamma_{i-h}^{0,1} \cdot \Gamma^{i-h,j-1}. \quad \square \end{aligned}$$

根据引理1和引理2,可以进一步得到定理1.

证明 记  $\|\Gamma_*^{0,1}\| = \max \|\Gamma_t^{0,1}\|$ , 则由引理1有

$$\|\Gamma^{i,j}\| \leq \sum_{k=0}^j \|\Gamma_*^{0,1}\|^k \|\Gamma^{1,0}\| \|\Gamma^{i-1,j-k}\|,$$

从而

$$\begin{aligned} & \sum_{j=0}^{\infty} \|\Gamma^{i,j}\| \leq \\ & \sum_{h=0}^{\infty} \|\Gamma_*^{0,1}\|^h \|\Gamma^{1,0}\| \sum_{j=0}^{\infty} \|\Gamma^{i-1,j}\| \leq \\ & \left( \sum_{h=0}^{\infty} \|\Gamma_*^{0,1}\|^h \|\Gamma^{1,0}\| \right)^2 \sum_{j=0}^{\infty} \|\Gamma^{i-2,j}\| \leq \\ & \left( \sum_{h=0}^{\infty} \|\Gamma_*^{0,1}\|^h \|\Gamma^{1,0}\| \right)^l \sum_{j=0}^{\infty} \|\Gamma^{i-l,j}\| \leq \\ & \left( \sum_{h=0}^{\infty} \|\Gamma_*^{0,1}\|^h \|\Gamma^{1,0}\| \right)^i \sum_{j=0}^{\infty} \|\Gamma^{0,j}\| \leq \\ & \left( \sum_{h=0}^{\infty} \|\Gamma_*^{0,1}\|^h \|\Gamma^{1,0}\| \right)^i \sum_{j=0}^{\infty} \|\Gamma_*^{0,1}\|^j. \end{aligned}$$

根据  $\|\Gamma_*^{0,1}\| < 1$ , 有

$$\sum_{j=0}^{\infty} \|\Gamma^{i,j}\| \leq \|\Gamma^{1,0}\|^i (1 - \|\Gamma_*^{0,1}\|)^{-i+1}.$$

因为  $i < N$ , 上式右边为一个有限的数, 因此定理1得证.  $\square$

定理1说明, 在只有  $\lambda(t)$  引入的情况下, 只要保证所有的  $\Gamma_t^{0,1}$  中最大的模小于1, 就可以保证在批次趋于无穷时状态转移矩阵趋于0.

两个引理的证明过程并不仅限于式(11)中2D系统所描述的形式, 对于其他有时变参数的2D系统的状态转移矩阵同样适用; 另一方面, 非时变的状态转移矩阵是时变的一个特例, 因此, 可以把非时变系统统一到该表述中. 虽然一定程度上增加了计算复杂度, 但可以得到更一般的2D系统状态转移特性.

### 2.2 状态预测与控制量计算

文献[18]中给出了2D-IPILC详细的状态预测和控制量计算方法, 本文仅仅列出有关的步骤.

在第  $k$  批次内, 取预测步长为  $m$ , 根据系统的全响应公式(17), 系统的预测输出为

$$\begin{aligned} \phi(t+m, k) = & \\ & \sum_{i=0}^{t+m} \Gamma^{t+m-i, k} \bar{e}(i, 0) + \\ & \sum_{(0,0) \leq (i,j) \leq (t+m,k)} (\Gamma^{t+m-i-2, k-j+1} \bar{B} + \\ & \Gamma^{t+m-i-1, k-j} \bar{D}) \Delta u_{\text{mpc}}(i, j). \quad (20) \end{aligned}$$

根据预测步长  $m$ , 将时刻  $t$  到  $t+m-1$  的预测输出写成如下向量形式<sup>[19]</sup>:

$$\hat{\Xi}_{(t+m)}^{(t+1)} | t, k = \Phi_{(t+m)}^{(t+1)} | t, k + G_t \Delta U_{\text{mpc}}(t_{t+m-1} | t, k). \quad (21)$$

其中

$$\begin{aligned} \hat{\Xi}_{(t+m)}^{(t+1)} | t, k &= \begin{bmatrix} \hat{\xi}(t+1 | t, k) \\ \vdots \\ \hat{\xi}(t+m | t, k) \end{bmatrix}, \\ \Phi_{(t+m)}^{(t+1)} | t, k &= \begin{bmatrix} \phi(t+1, k) \\ \vdots \\ \phi(t+m, k) \end{bmatrix}, \\ \Delta U_{\text{mpc}}(t_{t+m-1} | t, k) &= \begin{bmatrix} \Delta u_{\text{mpc}}(t, k) \\ \vdots \\ \Delta u_{\text{mpc}}(t+m-1, k) \end{bmatrix}. \end{aligned}$$

式(21)中等号右边第1部分为在第  $k$  批次  $t$  时刻根据已知信息可以获得的状态信息, 第2部分为根据第  $k$  批次  $t$  时刻之前的信息进行预测得到的预测控制输入部分对状态信息的影响.

根据该预测值可以计算2D-IPLC的输入. 考虑如下形式的目标函数:

$$J_m = \hat{\Xi}^{(t+1)}(t, k)^T Q \hat{\Xi}^{(t+1)}(t, k) + \Delta U_{\text{mpc}}(t, k)^T R \Delta U_{\text{mpc}}(t, k). \quad (22)$$

其中

$$Q = \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_m \end{bmatrix}_{(n+1) \cdot m \times (n+1) \cdot m},$$

$$Q_i = \begin{bmatrix} \beta_1 & 0 & \cdots & 0 & 0 \\ 0 & \beta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \beta_i & 0 \\ 0 & 0 & \cdots & 0 & \alpha_i \end{bmatrix}_{(n+1) \times (n+1)},$$

$$R = \begin{bmatrix} \gamma_1 & 0 & \cdots & 0 \\ 0 & \gamma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_m \end{bmatrix}_{m \times m},$$

且参数满足条件  $\alpha_i > 0, \beta_i \geq 0, \gamma_i \geq 0$ . 各权重系数分别用于最优化误差、状态变化和输入变化. 在无约束情况下对该目标函数进行求解, 可以得到其输入

$$\Delta U_{\text{mpc}}(t, k) = -[G_t^T Q G_t + R]^{-1} G_t^T Q \Phi^{(t+1)}(t, k). \quad (23)$$

其中

$$G_t = \begin{bmatrix} g_{t,t} \\ \vdots \\ g_{t+m,t} \cdots g_{t+m,t+m} \end{bmatrix}_{(n+1) \cdot m \times m},$$

$$g_{r,s} = \Gamma^{r-s-1, \lambda+1} \bar{B} + \Gamma^{r-s, \delta} \bar{D}, \quad \delta = i - j.$$

综上, 轨迹更新的2D-IPILC算法的基本步骤可以总结如下.

**Step 1:** 根据跟踪点选定一条初始跟踪轨迹, 使得该条初始轨迹经过指定的跟踪点, 并根据收敛条件选定其中的参数  $L$ .

**Step 2:** 根据式(5)选定满足轨迹更新条件的参数  $\lambda(t)$ .

**Step 3:** 根据P-ILC的算法  $\Delta u_{\text{ILC}} = L \cdot e(t, k)$  确定批次间的输入部分.

**Step 4:** 根据式(20)计算MPC的预测输入部分.

**Step 5:** 根据式(23), 取其第1项确定输入.

**Step 6:** 如果  $t < N$ , 则令  $t = t + 1$ , 并转至Step 4; 如果  $t = N$ , 则令  $k = k + 1$ , 转至Step 3.

上述算法中, 参数矩阵  $Q$  和  $\lambda(t)$  的选择会影响系统的收敛性, 预测步长  $m$  的选择会影响系统的收敛速度.

### 3 收敛性分析

算法收敛是指当批次趋于无穷时, 在特定点集  $M$  上的点对点跟踪误差  $e_{kM}$  满足  $\lim_{k \rightarrow \infty} e_{kM} = 0$ , 即意味着  $\lim_{k \rightarrow \infty} e(t, k) = 0, t \in M$ . 在2D系统中, 要保证算法收敛, 即要求满足  $\lim_{j \rightarrow \infty} \xi(i, j) = 0, i \in M$ .

本文首先考虑简单情况,  $\Delta d(i, j) = 0$  时系统的收敛情况. 记

$$\Gamma_{\Delta u_{\text{mpc}}}(i, j) = \Gamma^{t+m-i-2, k-j+1} \bar{B} + \Gamma^{t+m-i-1, k-j} \bar{D}.$$

根据系统的全响应公式得到

$$\xi(t+m, k) = \sum_{i=0}^t \Gamma^{t-i, k} \bar{e}(i, 0) + \sum_{(0,0) \leq (i,j) < (t+m,k)} \Gamma_{\Delta u_{\text{mpc}}}(i, j) \Delta u_{\text{mpc}}(i, j) = \Gamma_t^{1,0} \sum_{i=0}^t \Gamma^{t-i, k-1} \bar{e}(i, 0) + \Gamma_t^{1,0} \sum_{(0,0) \leq (i,j) < (t+m, k-1)} \Gamma_{\Delta u_{\text{mpc}}}(i, j) \Delta u_{\text{mpc}}(i, j) + \sum_{(0,k) \leq (i,j) < (t,k)} \Gamma_{\Delta u_{\text{mpc}}}(i, j) \Delta u_{\text{mpc}}(i, j) + \sum_{(t,k) \leq (i,j) < (t+m,k)} \Gamma_{\Delta u_{\text{mpc}}}(i, j) \Delta u_{\text{mpc}}(i, j).$$

由上式可知, 全响应包括4部分, 第1部分和第2部分可根据上一个批次的信息得到, 第3部分可根据  $k$  批次  $t$  时刻以前的信息得到, 第4部分为预测部分.

根据定理1, 响应的第1部分

$$\lim_{k \rightarrow \infty} \sum_{i=0}^t \Gamma^{t-i, k} \bar{e}(i, 0) = 0. \quad (24)$$

因此, 进一步根据预测输入, 在  $k$  批次  $t$  时刻对于之后  $m$  时刻的状态信息的预测中, 由已知信息确定的部分可以写成

$$\Phi_{(t+m)}^{(t+1)}(t, k) = \begin{bmatrix} \phi(t+1, k) \\ \vdots \\ \phi(t+m, k) \end{bmatrix} = f_B(t+1) \Xi + \bar{G}_t \Delta U_{\text{mpc}}(t, k), \quad (25)$$

其中

$$f_{B(t+m)} = \begin{bmatrix} \Gamma_{t+1}^{0,1} \\ \vdots \\ \Gamma_{t+m}^{0,1} \end{bmatrix},$$

$$\bar{G}_t = \begin{bmatrix} \Gamma^{t-1,1} \cdot \bar{B} + \Gamma^{t,0} \cdot \bar{D} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \Gamma^{t+m-1,1} \cdot \bar{B} + \Gamma^{t+m,0} \cdot \bar{D} & \cdots & 0 \end{bmatrix},$$

$$\Delta U_{\text{mpc}}(t, k) = \begin{bmatrix} \Delta u_{\text{mpc}}(0, k) \\ \vdots \\ \Delta u_{\text{mpc}}(t-1, k) \end{bmatrix},$$

$$\Xi_{(t+m, k-1)} = \begin{bmatrix} \xi(t+1, k-1) \\ \vdots \\ \xi(t+m, k-1) \end{bmatrix}.$$

将式(22)代入(20),将预测输入从0计算到  $N-1$ ,并写出向量形式.同时,记  $H_t \triangleq -[G_t^T Q G_t + R]^{-1} G_t^T Q$ ,则可以得到

$$\Delta U_{\text{mpc}}(k) = \begin{bmatrix} \Delta u_{\text{mpc}}(0, k) \\ \vdots \\ \Delta u_{\text{mpc}}(N-1, k) \end{bmatrix} =$$

$$\begin{bmatrix} H_0 \bar{I}_0 f_{BN} \Xi(k-1) \\ \vdots \\ H_{N-1} \bar{I}_{N-1} f_{BN} \Xi(k-1) \end{bmatrix} +$$

$$\begin{bmatrix} H_0 \bar{G}_t \tilde{I}_0 \Delta U_{\text{mpc}}(k) \\ \vdots \\ H_{N-1} \bar{G}_{N-1} \tilde{I}_{N-1} \Delta U_{\text{mpc}}(k) \end{bmatrix} =$$

$$H_E f_{BN} \Xi(k-1) + G_E \Delta U_{\text{mpc}}(k). \quad (26)$$

其中

$$\bar{I}_t = [0, I_1, 0]_N, \tilde{I}_t = [I_t, 0]_{N-t},$$

$$H_E = \begin{bmatrix} H_0 \bar{I}_0 & 0 \\ & \ddots \\ 0 & H_{N-1} \bar{I}_{N-1} \end{bmatrix},$$

$$G_E = \begin{bmatrix} G_0 \tilde{I}_0 & 0 \\ & \ddots \\ 0 & G_{N-1} \tilde{I}_{N-1} \end{bmatrix},$$

$$f_{BN} \triangleq f_{B(N-1)},$$

$$\Xi(k-1) \triangleq \Xi_{N-1}^0, k-1.$$

需要说明的是,虽然每次的预测步长为  $m$ ,但每次计算得到  $m$  步预测输入时,在当前时刻只取其预测值的第1项作为输入,引入时间轴的反馈控制

中.由于  $G_E$  为对角阵,且每一项对角元素不为0,可以得到预测输入和上一批次状态之间的关系为

$$\Delta U_{\text{mpc}}(k) = (I - G_E)^{-1} \cdot H_E \cdot f_{BN} \Xi_{k-1}. \quad (27)$$

将式(27)代入全响应公式(17),有

$$\Xi_k =$$

$$f_{BN} \Xi_{k-1} + G_N \cdot \Delta U_{\text{mpc}}(k) =$$

$$f_{BN} \Xi_{k-1} + G_N (I - G_E)^{-1} \cdot H_E \cdot f_{BN} \Xi_{k-1} =$$

$$(I + G_N (I - G_E)^{-1} \cdot H_E) f_{BN} \Xi_{k-1}. \quad (28)$$

从而得到收敛性条件

$$\|(I + G_N (I - G_E)^{-1} \cdot H_E) f_{BN}\| < \delta \leq 1. \quad (29)$$

式(29)中给出的收敛条件保证了  $\lim_{k \rightarrow \infty} \Xi_k = 0$ ,也意味着  $\lim_{k \rightarrow \infty} e_k = 0$ ,根据式(7)可得

$$\lim_{k \rightarrow \infty} e_{kM} = \Psi e_k = 0. \quad (30)$$

对比一般的收敛性条件,  $\|\Gamma^{0,1}\| = \|I - \text{CBL}\| < 1$ ,在点对点的综合预测迭代学习控制中,因为  $f_{BN}$  中包含了批次间更新轨迹参数  $\lambda_k(t)$ ,可以发现其参数选择范围更宽,例如取

$$\begin{cases} \|f_{BN}\| = \max \|\Gamma_t^{0,1}\| < 1, \\ \|(I + G_N (I - G_E)^{-1} \cdot H_E)\| < 1. \end{cases} \quad (31)$$

式(31)给出的条件只是(29)的一个充分条件,并不是必要条件.而在没有该轨迹更新和点对点问题的情况下,式(31)或其等价条件通常被认为是必要的.

参数范围选择更广,使得在批次间比例学习率部分的  $L$  选择范围更广.由于在批次间的学习率部分,输入与  $L$  紧密相关,在输入受限制的情况下,  $L$  的选择范围更广使得输入的可行集更大.

同时,根据轨迹更新律(6),可以得到

$$r_{k+1} = r_k + \lambda_k(r_k - y_k) = r_k + \lambda_k e_k. \quad (32)$$

因为  $e_k \rightarrow 0, k \rightarrow \infty$ ,所以  $\|r_k - r_{k-1}\| \rightarrow 0, k \rightarrow \infty$ .即

$$\lim_{k \rightarrow \infty} r_k = r^*. \quad (33)$$

上式说明跟踪轨迹最后会收敛到一条满足特定跟踪点目标的轨迹上,且与初始轨迹的选择无关,因此无需过多的先验信息来设计一条最优的初始跟踪轨迹.

当系统满足式(29)的收敛条件,且过程干扰的边界满足  $\Delta d(k) = [\Delta d(1, k), \Delta d(2, k), \dots, \Delta d(N, k)]^T \leq B_d$ ,其中  $B_d$  为边界.因此,系统的跟踪误差将收敛到零附近的一个邻域内,邻域边界为  $\sqrt{B_d^*/(1-\sigma)}$ ,其中

$$B_d^* = (1 + \mu) B_d^2 \|G_d^T G_d\|, 0 < \sigma < 1, \mu > 0. \quad (34)$$

具体的证明同样可以利用系统的全响应公式并借助三角不等式证明, 本文限于篇幅不详细。

### 4 仿真分析

采用典型机械臂模型<sup>[15]</sup>来对比分析基于轨迹更新的2D算法和Freeman提出的梯度下降算法以及固定轨迹算法。在固定轨迹算法中, 当初始轨迹选好后, 为了更好地比较算法效果, 选择的也是IPILC算法。仿真对象模型为一个非最小相位系统, 其连续传递函数模型<sup>[15]</sup>为

$$G(s) = \frac{165.95(4 - s)}{s^4 + 21.5s^3 + 170.28s^2 + 368.52s + 663.82}$$

设定采样周期为1s, 则可求解得到系统的离散时间状态空间模型为

$$A = \begin{bmatrix} 0.0198 & 0.4023 & 2.8250 & 2.5339 \\ -0.0038 & -0.623 & -0.2477 & 1.4183 \\ -0.0021 & -0.0498 & -0.4216 & -1.0351 \\ 0.0016 & 0.0314 & 0.2158 & 0.1485 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.0038 \\ -0.0021 \\ 0.0016 \\ 0.0013 \end{bmatrix},$$

$$C = [0 \ 0 \ -165.95 \ 663.80].$$

设定批次周期为  $T = 100\text{s}$ , 即  $N = 100$ . 跟踪目标轨迹为  $y_{dM} = [6, 8, -5, 9, 10]^T$ , 相应的特定时间点集为  $M = \{32, 40, 65, 92, 100\}$ . 过程干扰取峰值为0.4的随机噪声. 2D-IPILC的相关参数选择为比例学习率  $L = 0.05$ ,  $\alpha = 1$ ,  $\beta = 1$ ,  $\gamma = 1$ ,  $\lambda(t) = -0.65$ ,  $t \in M$ .

采用如下的跟踪误差平方和来判断方法的性能:

$$SSE_k = \sum \|y_d(t) - y_k(t)\|, t \in M. \quad (35)$$

为了与固定轨迹算法进行比较, 固定轨迹和轨迹更新算法设计一样的初始参考轨迹. 给出了两种初始轨迹方案: 方案1为一条平滑的类正弦的跟踪曲线, 方案2为除在关键点其余输入均为0的跟踪轨迹. 图1给出了关键点和两种初始跟踪轨迹的方案。

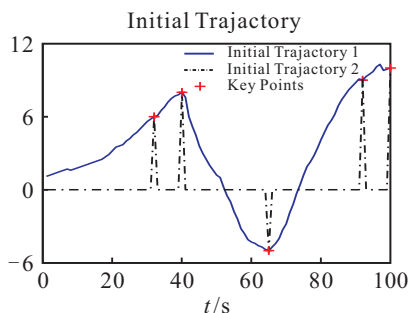


图1 两种初始轨迹设计

在方案1的初始轨迹情况下进行仿真, 得到图2的跟踪误差曲线. 由图2可知: 与固定参考轨迹相比, 轨迹更新的算法收敛速度更快; 与梯度下降算法相比, 2D-IPILC算法可以更好地抑制过程干扰. 因此本文算法在收敛速度和鲁棒性上都有相对更好的表现。

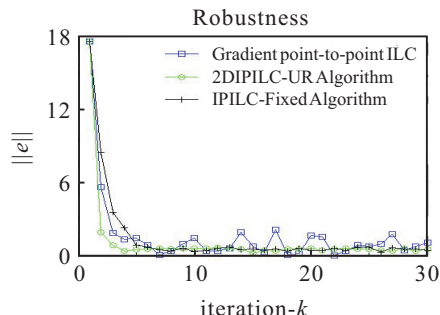


图2 与梯度下降和固定轨迹算法的比较

为了进一步说明式(33)的特性, 在方案2的情况下对3种算法的跟踪误差进行了比较。

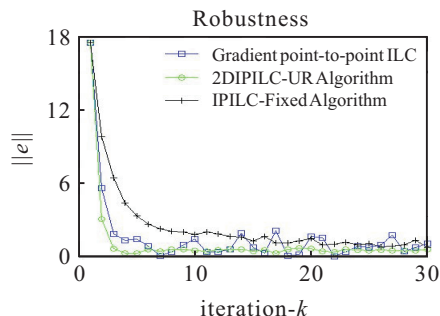


图3 在初始轨迹2下与固定轨迹比较

从图3可以发现: 在方案2下, 固定轨迹的效果明显变差, 而轨迹更新算法则不受影响, 同时与优化算法相比, 仍然保持了其优点, 收敛速度相当, 抗干扰能力更好; 而与固定轨迹算法相比, 则在两个方面都有明显的优势。

同时, 对于该系统可以发现, 在固定轨迹的情况下, 当  $L = 1$  时, 该算法是发散的. 检验其收敛性条件  $\|f_B(I + G_N(I - G_E)^{-1}H_E)\| = 2.0359$ , 不满足收敛性条件; 而在轨迹更新的情况下, 当  $L = 1$  时, 系统仍然可以收敛, 此时  $\|f_B(I + G_N(I - G_E)^{-1}H_E)\| = 0.9936$ , 满足收敛性条件. 因此, 本文提出的算法提供了参数  $L$  更大的选择范围. 对于可调节的参数  $\lambda(t)$  而言, 在满足收敛性条件的情况下,  $\|1 + \lambda(t)\|$  越大, 收敛速度越快. 图4给出了在  $L = 0.02$  时,  $\lambda(t)$  的取值在  $[-2, 0]$  变化时系统的收敛情况. 图中箭头所指方向为  $\|1 + \lambda(t)\|$  减小的方向, 可知随着  $\|1 + \lambda(t)\|$  的减小, 系统可以在更少的批次内收敛. 虽然让轨迹更新的幅度变大可以使得收敛速度更快, 但跟踪轨迹变化过大也可能使得系统稳定性变差。

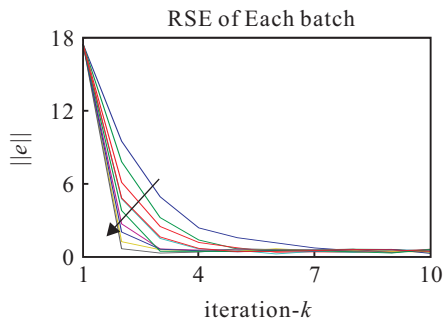


图4  $\|1 + \lambda(t)\|$ 的影响

## 5 结论

本文基于2D理论,针对点对点的跟踪问题,提出了一种结合轨迹更新的综合预测迭代学习控制方法.采用2D理论详细分析了在目标轨迹更新的策略下,含有时变参数的2D系统状态转移矩阵的性质以及系统的全响应表达式,并基于全响应公式给出了该控制算法的收敛性条件.与优化问题相比,该算法具有更好的鲁棒性;同时,与固定轨迹的方法相比,该算法具有更快的收敛速度.此算法充分利用了点对点问题在非特定点的自由度,使得算法在迭代学习控制律各参数的选择范围更大.最后,分析了参数选择对算法性能的影响.

## 参考文献(References)

- [1] Bristow D A, Tharayil M, Alleyne A G. A survey of iterative learning control[J]. *IEEE Control Systems*, 2006, 26(3): 96-114.
- [2] Wang Y, Gao F, Doyle F J. Survey on iterative learning control, repetitive control, and run-to-run control[J]. *J of Process Control*, 2009, 19(10): 1589-1600.
- [3] Alhazza K A, Hasan A M, Alghanim K A, et al. An iterative learning control technique for point-to-point maneuvers applied on an overhead crane[J]. *Shock and Vibration*, 2014, 2014: 261-509.
- [4] Freeman C T, Dinh T V. Experimentally verified pointto-point iterative learning control for highly coupled systems[J]. *Int J of Adaptive Control and Signal Processing*, 2015, 29(3): 302-324.
- [5] Freeman C T, Cai Z, Rogers E, et al. Iterative learning control for multiple point-to-point tracking application[J]. *IEEE Trans on Control Systems Technology*, 2011, 19(3): 590-600.
- [6] Ito Y, Park Y S, Imanishi Y. Visualization of critical pH-controlled gating of a porous membrane grafted with polyelectrolyte brushes[J]. *J of the American Chemical Society*, 1997, 119(11): 2739-2740.
- [7] Xu J X, Chen Y, Lee T H, et al. Terminal iterative learning control with an application to RTPCVD thickness control[J]. *Automatica*, 1999, 35(9): 1535-1542.
- [8] Hou Z S, Wang Y, Yin C K, et al. Terminal iterative learning control for station stop control of a train[C]. *Int J of Control*, 2011, 84(7): 1263-1274.
- [9] Park J, Chang P H, Park H S, et al. Design of learning input shaping technique for residual vibration suppression in an industrial robot[J]. *IEEE/ASME Trans on Mechatronics*, 2006, 11(1): 55-65.
- [10] Singer N C, Seering W P. Preshaping command inputs to reduce system vibration[J]. *J of Dynamic Systems, Measurement, and Control*, 1990, 112(1): 76-82.
- [11] Dijkstra B G, Rambaratsingh N J, Scherer C, et al. Input design for optimal discrete time point-to-point motion of an industrial XY-positioning table[C]. *Proc of the 39th IEEE Conf on Decision and Control*. Sydney: IEEE, 2000: 901-906.
- [12] Freeman C T. Constrained point-to-point iterative learning control with experimental verification[J]. *Control Engineering Practice*, 2012, 20(5): 489-498.
- [13] Chu B, Freeman C T, Owens D H. A novel design framework for point-to-point ILC using successive projection[J]. *IEEE Trans on Control Systems Technology*, 2015, 23(3): 1156-1163.
- [14] Owens D H, Freeman C T, Van Dinh T. Norm-optimal iterative learning control with intermediate point weighting: Theory, algorithms, and experimental evaluation[J]. *IEEE Trans on Control Systems Technology*, 2013, 21(3): 999-1007.
- [15] Freeman C T, Tan Y. Iterative learning control with mixed constraints for point-to-point tracking[J]. *IEEE Trans on Control Systems Technology*, 2013, 21(3): 604-616.
- [16] 安通鉴, 刘祥官. 目标轨迹更新的点到点鲁棒迭代学习控制[J]. *浙江大学学报: 工学版*, 2015, 49(1): 87-92. (An T J, Liu X G. Point-to-point robust iterative learning control via reference trajectory regulating[J]. *J of Zhejiang University: Engineering Science*, 2015, 49(1): 87-92.)
- [17] Freeman C T, Cai Z, Lewin P L, et al. Iterative learning control for multiple point-to-point tracking[C]. *Proc of the 48th IEEE Conf on Decision and Control*. Shanghai: IEEE, 2009: 3288-3293.
- [18] Chen C, Xiong Z, Zhong Y. Design and analysis of integrated predictive iterative learning control for batch process based on two-dimensional system theory[J]. *Chinese J of Chemical Engineering*, 2014, 22(7): 762-768.
- [19] Qiu W, Ma J, Xiong Z, et al. A robust integrated predictive iterative learning control based on updating reference for point-to-point tracking[C]. *American Control Conf*. Boston: IEEE, 2016: 1265-1270.
- [20] Tayebi A. Adaptive iterative learning control for robot manipulators[J]. *Automatica*, 2004, 40(7): 1195-1203.
- [21] Roesser R P. A discrete state-space model for linear image processing[J]. *IEEE Trans on Automatic Control*, 1975, 20(1): 1-10.
- [22] Lee J H, Lee K S. Iterative learning control applied to batch processes: An overview[J]. *Control Engineering Practice*, 2007, 15(10): 1306-1318.