

# 带有初态误差的高阶多智能体系统一致性跟踪

李国军<sup>†</sup>

(浙江工业大学信息工程学院, 杭州 310023)

(浙江警察学院公共基础部, 杭州 310053)

(E-mail: liguojun@zjjcxy.cn)

陈东杰 韩一士

(浙江警察学院公共基础部, 杭州 310053)

(E-mail: chendongjie@zjjcxy.cn; hanyishi@zjjcxy.cn)

**摘要** 本文借助迭代学习控制方法, 针对一类在有限区间上执行重复任务的高阶多智能体, 提出一种一致性跟踪算法。在跟踪过程中, 通过对状态误差的修正, 实现了一致性跟踪。在修正过程中, 系统在某段时间内只修正某一种初态误差, 当这种初态误差的修正操作完成以后紧接着开始下一种初态误差的修正, 以此类推, 最终实现所有初态误差的完全修正, 并且所有修正操作在一个指定的时间内完成。最后, 通过仿真算例验证了所提算法的有效性。

**关键词** 多智能体; 迭代学习控制; 有向图; 收敛; Step-by-step 修正

**MR(2000) 主题分类** 93E35

**中图分类** O212.7

## 1 引言

近年来, 由于多智能体在无人机控制, 机器人编队控制等方面的应用, 多智能体系统的研究已经成为控制领域的一个研究热点<sup>[1,2]</sup>, 引起了控制, 数学, 物理, 计算机以及人工智能等众多领域专家的兴趣<sup>[3]</sup>。在对多智能体的控制过程中, 一致性显得尤为重要。所谓一致性, 是指多智能体系统中的各个单智能体通过相互间的协作能够达到同一个状态或者同一个输出<sup>[4,5]</sup>。

本文 2017 年 5 月 10 日收到, 2017 年 12 月 12 日收到修改稿。

<sup>†</sup> 通讯作者。

21世纪初, 大量的研究人员尝试着用 Lyapunov 方法或压缩映射方法来解决多智能体的一致性问题<sup>[6,7]</sup>. 早期的文献主要关注的都是低维系统<sup>[8]</sup>, 目前, 越来越多的学者开始关注高阶高维多智能体系统<sup>[9–11]</sup>. [12] 对具有强连通子图拓扑结构的有向多智能体系统实现了网络的可控包含控制. [13] 研究了有领导者的线性多智能体系统的一致性问题, 并将其应用于编队控制. [14] 研究了多智能体网络系统在连续和离散两种状态下的一致性, 并证明了两种状态下一致平衡点的凸组合系数相等. [15] 利用多层递阶结构设计算法, 有效加快了多智能体系统一致性的收敛速度. 相比较于线性多智能体而言, 非线性多智能体更难控制, [16] 和 [17] 则是借助神经网络和模糊系统的方法实现了对非线性多智能体的一致性控制.

迭代学习控制 (Iterative learning control, 简称 ILC) 适用于在有限区间作重复操作的场合, 其控制思想是沿着迭代轴利用系统跟踪误差不断修正控制输入, 改进输出效果, 经过多次迭代以后, 系统能够达到任意小误差跟踪<sup>[17,18]</sup>. 由于设计简单, 在线计算量小, 甚至在不需要知道系统具体模型的情况下就能实现对系统的控制, 因此在机器人控制, 化工过程控制以及产品线加工过程控制等场合得到应用. 并且迭代学习控制能够实现零误差跟踪, 因此一些学者开始运用迭代学习控制的方法来解决多智能体的一致性跟踪问题. 文献 [19–20] 较早地利用迭代学习控制方法解决了多智能体系统的一致性跟踪问题.

在应用 ILC 方法时, 要求每次的初始定位误差等于零, 但这在实际中不可能做到, 因此这种严格的要求也限制了 ILC 的应用. 为了在有初态误差的条件下实现在指定区间上的无误差跟踪, 很多学者都做了大量有益的工作<sup>[22–32]</sup>, 并得到一些有益的结论. [22] 利用迭代学习控制方法实现了离散和连续系统的一致性跟踪, 并用初始修正方法解决了初始状态误差, 但此方法不适用于高阶多智能体系统. [31] 利用初态学习的方法, 实现了对异质多智能体的一致性跟踪. 针对一类带有任意误差的主从型非参数化不确定多智能体系统, [32] 采用鲁棒策略和学习策略, 实现了多智能体的一致性跟踪, 但文中方法也不能解决高阶多智能体的一致性跟踪问题.

本文针对带有初始状态误差的高阶多智能体系统, 通过 step-by-step 修正方法修正状态误差, 并借助高阶导数信号实现了多智能体的一致性跟踪. 与已有参考文献相比, 本文算法不需要进行初态学习, 因为在实际控制过程中, 并不是每次迭代都能达到学习的要求. 而且该算法体现了一种新的控制思想即 step-by-step 控制方法. 最后, 通过数字仿真验证了该算法的有效性.

## 2 问题阐述

考虑下面的多智能体系统, 且该系统包含  $m_a$  个智能体.

$$\begin{cases} \dot{x}_{j,k}^i(t) = x_{j,k}^{i+1}(t), & i = 1, 2, \dots, n-1, \\ \dot{x}_{j,k}^n(t) = \mathbf{A}x_{j,k}^n(t) + \mathbf{B}u_{j,k}(t), \\ y_{j,k}(t) = \mathbf{C}x_{j,k}^1(t), \end{cases} \quad (1)$$

其中,  $t \in [0, T]$  是有限时间;  $\mathbf{A} \in R^{m_s \times m_s}$ ;  $\mathbf{B} \in R^{m_s \times r}$  ( $r \geq m_s$ );  $\mathbf{C} \in R^{s \times m_s}$ ;  $k = 1, 2, \dots$  表示迭代次数;  $x_{j,k}^i(t) \in R^{m_s}$  ( $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m_a$ ) 表示第  $j$  个智能体第  $k$  次迭代的状态;  $u_{j,k}(t) \in R^r$ ;  $y_{j,k}(t) \in R^s$  分别是第  $j$  个智能体的输入向量和输出向量 (前述变量相应可以简写为  $x_{j,k}^i, u_{j,k}, y_{j,k}$ ).

设  $y_d(t)$  (简写为  $y_d$ ) 是给定的期望轨迹, 设  $x_d^i(t)$  (相应简写为  $x_d^i$ ) 是给定的期望状态,  $e_{j,k}(t) = y_d - y_{j,k}$  表示第  $j$  个智能体的第  $k$  次输出误差 ( $e_{j,k}(t)$  简写为  $e_{j,k}$ ).

定义

$$e_{j_1,j_2,k}(t) = y_{j_1,k} - y_{j_2,k} = (y_d - y_{j_2,k}) - (y_d - y_{j_1,k}) = e_{j_2,k} - e_{j_1,k}. \quad (2)$$

从上面的定义可知,  $e_{j_1,j_2,k}(t)$  (简写为  $e_{j_1,j_2,k}$ ) 表示  $j_1$  个智能体和第  $j_2$  个智能体的相对误差. 如果两个智能体相邻, 那么在多智能体运行过程中,  $e_{j_1,j_2,k}$  对实现多智能体的一致性跟踪将起到重要作用. 另外, 多智能体系统要求至少有一个智能体能够接收到期望轨迹  $y_d$  的信号.

多智能体系统的  $m_a$  个智能体构成一个网络拓扑图. 假定  $G = (V, \Xi, \Lambda)$  表示此有向图.  $V(G)$  是顶点集并且  $V(G) = \{v_1, v_2, \dots, v_{m_a}\}$ , 其中  $v_j$  表示第  $j$  个智能体.  $\Xi(G)$  是边集合, 如果第  $j_1$  个智能体能够直接从第  $j_2$  个智能体接收到信号, 那么  $(v_{j_1}, v_{j_2}) \in \Xi(G) \subseteq V(G) \times V(G)$ .  $\mathbb{N}_j = \{j : (v_j, v_i) \in \Xi(G)\}$  表示顶点  $v_j$  的邻接点集合. 图  $G$  的邻接矩阵  $\Lambda = [a_{ij}]$  按如下方式定义: 如果  $(v_{j_1}, v_{j_2}) \in \Xi(G)$ , 那么  $a_{ij} > 0$ ; 否则  $a_{ij} = 0$ ,  $a_{ij} > 0$  意味着智能体  $i$  和智能体  $j$  之间有信息交流, 并且通常情况下定义  $a_{ii} = 0$ . 图  $G$  的 Laplacian 矩阵  $L_G$  定义为:  $L_G = \Delta - \Lambda$ , 其中  $\Delta = \text{diag}\{\Delta_{11}, \Delta_{22}, \dots, \Delta_{m_a m_a}\}$  ( $\text{diag}$  表示对角阵), 且  $\Delta_{ii} = \sum_{j=1, j \neq i}^m a_{ij}$ . 如果有限的顶点序列  $v_{j_1}, v_{j_2}, \dots, v_{j_i}$  的任意两个相邻的顶点之间有一条边 (即  $(v_{j_{k-1}}, v_{j_k}) \in \Xi(G)$ ,  $k = 2, 3, \dots, i$ ), 则称此顶点序列为  $v_{j_1}$  到  $v_{j_i}$  的一条路径.

本文中的多智能体系统不仅包含  $m_a$  个智能体还包括一个虚拟领导者, 并且有部分智能体能够接直接收到虚拟领导者发出的信号. 这种情形下, 用矩阵  $S$  描述虚拟领导者和各单智能体之间的联系, 且  $S = \text{diag}\{s_1, s_2, \dots, s_{m_a}\}$ , 其中如果智能体  $i$  能够直接到虚拟领导者发出的信号, 那么  $s_i > 0$ , 否则  $s_i = 0$ .

**假设 1** 初始状态  $x_{j,k}^i(0)$  是某个固定值, 也就是说,  $e_{j,k}^{(i-1)}(0)$ ,  $i = 1, \dots, n$  在每次迭代时值相同, 并且状态  $x_{j,k}^i(t)$  可测.

### 3 控制器设计

本节的任务是设计一个控制算法, 使得多智能体系统的每一个智能体在指定时间内都能跟踪上期望轨迹. 为此, 我们提出下面的控制律.

$$\begin{aligned} u_{j,k+1}(t) = & u_{j,k}(t) + \Gamma \left[ s_j e_{j,k}^{(n)}(t) + a_{jr} \sum_{r \in \mathbb{N}_j} e_{r,j,k}^{(n)}(t) \right. \\ & \left. + s_j \Theta_n(t) e_{j,k}^{(n-1)}(0) + a_{jr} \Theta_n(t) e_{r,j,k}^{(n-1)}(0) \right] \end{aligned}$$

$$\begin{aligned}
& + s_j \sum_{i=1}^{n-1} \Theta_i(t) e_{j,k}^{(i-1)}(h_{i,1}) \\
& + a_{jr} \sum_{r \in \mathbb{N}_j} \sum_{i=1}^{n-1} \Theta_i(t) e_{r,j,k}^{(i-1)}(h_{i,1}),
\end{aligned} \tag{3}$$

式中,  $\Gamma$  为控制增益,  $\mathbb{N}_j$  表示智能体  $j$  的邻接点集合,  $s_j \in S$ ,  $a_{jr} \in \Lambda$  为图  $G$  的边增益, 并且

$$\begin{aligned}
h_{i,1} & = 2^{n-i-1} t_p, \\
\Theta_i(t) & = \begin{cases} 0, & t \in [0, h_{i,1}), \\ \overline{h_i} \theta_i, & t \in [h_{i,1}, h_{i,2}), \\ 0, & t \in [h_{i,2}, T], \end{cases} \\
\Theta_n(t) & = \begin{cases} \frac{6}{t_p^3} t(t_p - t), & t \in [0, t_p), \\ 0, & t \in [t_p, T), \end{cases}
\end{aligned} \tag{4}$$

其中,  $t_p$  是预先给定的时间常量,

$$\begin{aligned}
h_{i,2} & = 2 * h_{i,1}, \\
\overline{h_i} & = \frac{2}{(h_{i,1})^{4(N-i)+2}} \frac{[2(N-i)+1]!}{((N-i)!)^2}, \\
\theta_i & = [t^{N-i}(t - h_{i,1})^{2(N-i)+1}(h_{i,2} - t)^{N-i}]^{(n-i)}.
\end{aligned} \tag{5}$$

上式中的  $N > n$  为给定的正整数,  $(n-i)$  表示第  $n-i$  阶导数, 从上式中可以看出  $\theta_i$  是  $t^{N-i}(t - h_{i,1})^{2(N-i)+1}(h_{i,2} - t)^{N-i}$  的  $n-i$  阶导数.

控制律中的函数  $\Theta_i(t)$  非常特别, 关于  $\Theta_i(t)$ , 有下面的定理成立.

**定理 1** 对于任意一个  $\Theta_i(t)$ , 当积分上限满足  $t_0 \in [h_{i,2}, T]$  时, 有

$$\begin{aligned}
I_q & = \underbrace{\int_0^{t_0} \cdots \int_0^{t_{q-1}}}_{q} \Theta_i(t_q) dt_q \cdots dt_1 \\
& = \underbrace{\int_{h_{i,1}}^{h_{i,2}} \cdots \int_{h_{i,1}}^{t_{q-1}}}_{q} \Theta_i(t_q) dt_q \cdots dt_1 \\
& = \begin{cases} 1, & q = n - i + 1, \\ 0, & 0 < q < n - i + 1. \end{cases}
\end{aligned} \tag{6}$$

证 1)  $q = n - i + 1$ .

$$\begin{aligned} I_q &= \underbrace{\int_{h_{i,1}}^{h_{i,2}} \cdots \int_{h_{i,1}}^{t_{q-1}}}_{q} \Theta_i(t_q) dt_q \cdots dt_1 \\ &= \int_{h_{i,1}}^{h_{i,2}} \overline{h_i} t_1^{N-i} (t_1 - h_{i,1})^{2(N-i)+1} (h_{i,2} - t_1)^{N-i} dt_1. \end{aligned}$$

记  $\mu = t_1 - h_{i,1}$ , 则

$$\begin{aligned} I_q &= \int_0^{h_{i,1}} \overline{h_i} (\mu + h_{i,1})^{N-i} \mu^{2(N-i)+1} [h_{i,1} - \mu]^{N-i} d\mu \\ &= \frac{\overline{h_i}}{2} \int_0^{h_{i,1}} (h_{i,1}^2 - \mu^2)^{N-i} \mu^{2(N-i)} d(\mu^2) \\ &\stackrel{v=\mu^2, H=h_{i,1}^2}{=} \frac{\overline{h_i}}{2} \int_0^H (H - v)^{N-i} v^{N-i} dv \\ &\stackrel{v=w+\frac{H}{2}}{=} \frac{\overline{h_i}}{2} \int_{-\frac{H}{2}}^{\frac{H}{2}} \left( \left( \frac{H}{2} \right)^2 - w^2 \right)^{N-i} dw \\ &\stackrel{w=\frac{H}{2} \sin \alpha}{=} \overline{h_i} \left( \frac{H}{2} \right)^{2(N-i)+1} \int_0^{\frac{\pi}{2}} (\cos \alpha)^{2(N-i)+1} d\alpha. \end{aligned}$$

定义

$$I_{2(N-i)+1} = \int_0^{\frac{\pi}{2}} (\cos \alpha)^{2(N-i)+1} d\alpha,$$

则有,

$$\begin{aligned} I_{2(N-i)+1} &= \int_0^{\frac{\pi}{2}} (\cos \alpha)^{2(N-i)} d \sin \alpha \\ &= [\sin \alpha (\cos \alpha)^{2(N-i)}]_0^{\frac{\pi}{2}} - 2(N-i) \int_0^{\frac{\pi}{2}} (\cos \alpha)^{2(N-i)-1} \sin^2 \alpha d\alpha \\ &= -2(N-i) \int_0^{\frac{\pi}{2}} (\cos \alpha)^{2(N-i)-1} \sin^2 \alpha d\alpha \\ &= 2(N-i) I_{2(N-i)-1} - 2(N-i) I_{2(N-i)+1}. \end{aligned}$$

因此,

$$\begin{aligned} I_{2(N-i)+1} &= \frac{2(N-i)}{2(N-i)+1} I_{2(N-i)-1} \\ &= \frac{2(N-i)}{2(N-i)+1} \cdot \frac{2(N-i-1)}{2(N-i)-1} \cdots \frac{4}{5} \cdot \frac{2}{3}, \end{aligned}$$

所以有  $I_q = 1$ .

2)  $0 < q < n - i + 1$ .

这种情形下很容易得到下面的表达式.

$$I_q = \overline{h_i} [t^{N-i} (t - h_{i,1})^{2(N-i)+1} (h_{i,2} - t)^{N-i}]^{(n-i-q)}|_{h_{i,1}}^{h_{i,2}} = 0.$$

所以结论成立, 证毕.

**注 1** 这个定理表明, 每一个  $\Theta_i(t)$  函数, 在经过  $n-i+1$  次积分以后有类似于脉冲函数的作用. 在实际中, 脉冲函数是不存在的, 因此用这样一个函数来代替脉冲函数.

本文中称这种控制器为 step-by-step 修正控制器, 因为这种控制器在  $t_p$  时刻完成修正智能体  $j$  的状态  $x_{j,k}^n$  的误差, 在  $2^{(n-i)}t_p$  时刻完成修正  $x_{j,k}^i$  的误差, 以此类推, 在  $2^{(n-1)}t_p$  时刻修正好状态  $x_{j,k}^1$  的误差, 也就是说, 系统在  $2^{(n-1)}t_p$  时刻完全跟踪上期望轨迹.

为了便于后面的收敛性分析, 采用下面的形式来描述系统(1). 在这种描述方式下, 每一个智能体的状态方程都包含在其中.

$$\begin{cases} \dot{x}_k^i(t) = x_k^{i+1}(t), & i = 1, 2, \dots, n-1, \\ \dot{x}_k^n = (\mathbf{I}_{m_a} \otimes \mathbf{A})x_k^n + (\mathbf{I}_{m_a} \otimes \mathbf{B})u_k(t), \\ y_k(t) = (\mathbf{I}_{m_a} \otimes \mathbf{C})x_k^1(t), \end{cases} \quad (7)$$

其中,  $\mathbf{I}_{m_a}$  是  $m_a$  维的单位阵,  $x_k^i(t) = [(x_{1,k}^i(t))^T, (x_{2,k}^i(t))^T, \dots, (x_{m_a,k}^i(t))^T]^T \in R^{m_a * m_s}$ ,  $u_k(t) = [u_{1,k}^T(t), u_{2,k}^T(t), \dots, u_{m_a,k}^T(t)]^T \in R^{m_a * r}$ ,  $y_k(t) = [y_{1,k}^T(t), y_{2,k}^T(t), \dots, y_{m_a,k}^T(t)]^T \in R^{m_a * s}$ ,  $\otimes$  表示 Kronecker 积.

同时, 将控制律(3) 改成下面的描述方式

$$\begin{aligned} u_{k+1}(t) = & u_k(t) + (\mathbf{L} + \mathbf{S}) \otimes \Gamma e_k^{(n)}(t) \\ & + \Theta_n(t)(\mathbf{L} + \mathbf{S}) \otimes \Gamma e_k^{(n-1)}(0) \\ & + \sum_{i=1}^{n-1} \Theta_i(t)(\mathbf{L} + \mathbf{S}) \otimes \Gamma e_k^{(i-1)}(h_{i,1}). \end{aligned} \quad (8)$$

## 4 收敛性分析

这一节着重分析多智能体系统(7)在应用控制律(8)以后的收敛性, 在分析之前引入[24]中的定义和引理.

**定义 1** 函数  $x(t)$ ,  $t \in [0, T]$  的  $\lambda$  范数  $\|x(t)\|_\lambda$  按如下方式定义.

$$\|x(t)\|_\lambda = \sup_{t \in [0, T]} e^{-\lambda t} \|x(t)\|, \quad \lambda > 0, \quad (9)$$

其中,  $\|\cdot\|$  是按以下方式定义的某种范数. 如果  $Z(t)$  是一个  $n$  维向量并且  $Z(t) = (z_1(t), \dots, z_n(t))^T$ , 那么  $\|Z(t)\| = \max_{1 \leq i \leq n} |z_i(t)|$ ; 如果  $Z(t)$  是一个矩阵函数并且  $Z(t) = \{z_{ij}\} \in R^{m \times n}$ , 那么  $\|Z(t)\| = \max_{1 \leq i \leq m} (\sum_{j=1}^n |z_{ij}(t)|)$ . 关于  $\lambda$  范数, 有下面的引理.

**引理 1**

$$\begin{aligned} & \sup_{t \in [0, T]} \left( e^{-\lambda t} \underbrace{\int_0^{t_0} \cdots \int_0^{t_{p-1}}}_{p} \|Z(t_p)\| dt_p \cdots dt_1 \right) \\ & \leq \frac{1}{\lambda^p} \|Z(t_p)\|_\lambda. \end{aligned}$$

**引理 2** 设级数  $b_k$  满足下列条件,

$$b_k \leq \rho b_{k-1} + q.$$

如果  $0 \leq \rho < 1$ , 那么

$$\lim_{k \rightarrow \infty} |b_k| \leq \left| \frac{q}{1-\rho} \right|.$$

上节提出的控制律 (8) 中, 引入了初始修正函数  $\Theta_i(t)$ , 关于初始修正有下面的定理.

**定理 2** 如果初始状态是固定有限值, 并且

$$\|\mathbf{I}_{m_a m_s} - (\mathbf{L} + \mathbf{S}) \otimes \mathbf{C}B\Gamma\| < 1, \quad (10)$$

那么 step-by-step 修正控制律 (8) 能够让多智能体系统 (7) 跟踪上期望轨迹, 并且在区间  $t \in [2^{n-1}t_p, T]$  上, 能够达到一致性跟踪.

证 为了方便书写, 记

$$\begin{aligned} \Delta u_k(t) &= u_{k+1}(t) - u_k(t), \\ x_k^i &= x_k^i(t), \\ \Delta x_k^i &= x_{k+1}^i(t) - x_k^i(t), \\ y_{m,d}^{(i-1)}(t) &= \underbrace{(y_d^{(i-1)}(t)^T, y_d^{(i-1)}(t)^T, \dots, y_d^{(i-1)}(t)^T)^T}_{m_a}, \\ x_{m,d}^i(t) &= \underbrace{(x_d^i(t)^T, x_d^i(t)^T, \dots, x_d^i(t)^T)^T}_{m_a}. \end{aligned}$$

在控制律 (8) 的作用下, 当  $t \in [0, 2^{n-1}t_p]$ , 假定轨迹

$$\begin{aligned} y_{m,d}^{(n-1),*}(t) &= y_{m,d}^{(n-1)}(t) - (\mathbf{I}_{m_a} \otimes \mathbf{I}_1) e_k^{(n-1)}(0) \\ &+ \sum_{i=1}^{n-1} \int_0^t \Theta_i(t_1) (\mathbf{I}_{m_a} \otimes \mathbf{I}_1) e_k^{(i-1)}(h_{i,1}) dt_1 \\ &+ \int_0^t \Theta_n(t_1) (\mathbf{I}_{m_a} \otimes \mathbf{I}_1) e_k^{(n-1)}(0) dt_1 \end{aligned} \quad (11)$$

可实现 (其中,  $\mathbf{I}_1$  为合适维数的单位阵), 即当系统 (1) 存在固定初始偏差  $x_d^{(n-1),*}(0) \neq x_d^{(n-1)}(0)$  时, 存在控制量  $u_d^*(t)$  使得输出为  $y_{m,d}^{(n-1),*}(t)$ , 相应的误差记为  $e_k^{(n-1),*}(t) = y_{m,d}^{(n-1),*}(t) - y_k^{(n-1)}(t)$ .

应用控制律(8)能够得到

$$\begin{aligned}
& x_{k+1}^n(t) - x_k^n(t) \\
&= [\dot{x}_{k+1}^n(0) - \dot{x}_k^n(0)] + \int_0^t [\dot{x}_{k+1}^n(t) - \dot{x}_k^n(t)] dt_1 \\
&= \int_0^t (\mathbf{I}_{m_a} \otimes \mathbf{A})(x_{k+1}^n(t) - x_k^n(t)) dt_1 + \int_0^t (\mathbf{I}_{m_a} \otimes \mathbf{B})\Delta u_{i,k}(t_1) dt_1 \\
&= \int_0^t (\mathbf{I}_{m_a} \otimes \mathbf{A})(x_{k+1}^n - x_k^n) dt_1 + \int_0^t [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(n)}(t_1) dt_1 \\
&\quad + \int_0^t \Theta_n(t_1) [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(n-1)}(0) dt_1 \\
&\quad + \sum_{i=1}^{n-1} \int_0^t \Theta_i(t_1) [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(i-1)}(h_{i,1}) dt_1. \tag{12}
\end{aligned}$$

根据积分公式有

$$\begin{aligned}
& \int_0^t [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(n)}(t_1) dt_1 \\
&= [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(n-1)}(t) - [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(n-1)}(0). \tag{13}
\end{aligned}$$

将式(13)代入式(12), 可得

$$\begin{aligned}
& x_{k+1}^n(t) - x_k^n(t) \\
&= \int_0^t (\mathbf{I}_{m_a} \otimes \mathbf{A})(x_{k+1}^n - x_k^n) dt_1 + [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(n-1)}(t) \\
&\quad - [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(n-1)}(0) + \int_0^t \Theta_n(t_1) [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(n-1)}(0) dt_1 \\
&\quad + \sum_{i=1}^{n-1} \int_0^t \Theta_i(t_1) [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(i-1)}(h_{i,1}) dt_1.
\end{aligned}$$

在上式两端同时乘以矩阵  $\mathbf{C}$ , 由于  $e_k^{(n-1)}(t) = y_{m,d}^{(n-1)}(t) - y_k(t) = e_k^{(n-1),*}(t) + y_{m,d}^{(n-1)}(t) - y_{m,d}^{(n-1),*}(t)$ , 代入  $e_k^{(n-1)}(t)$ , 并化简可得

$$\begin{aligned}
e_{k+1}^{(n-1),*}(t) &= - \int_0^t (\mathbf{I}_{m_a} \otimes \mathbf{C}A)(x_{k+1}^n - x_k^n) dt_1 \\
&\quad + (\mathbf{I}_{m_a m_s} - [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{C}B\Gamma]) e_k^{(n-1),*}(t).
\end{aligned}$$

对上式取  $\lambda$  范数并化简可得

$$\begin{aligned}
& \|e_{k+1}^{(n-1),*}(t)\|_\lambda \\
&\leq \left\| - \int_0^t (\mathbf{I}_{m_a} \otimes \mathbf{C}A)(x_{k+1}^n - x_k^n) dt_1 \right\|_\lambda \\
&\quad + \|\mathbf{I}_{m_a m_s} - (\mathbf{L} + \mathbf{S}) \otimes \mathbf{C}B\Gamma\| \|e_k^{(n-1),*}(t)\|_\lambda.
\end{aligned}$$

对于  $\lambda$  范数, 如果  $\lambda$  足够大, 根据引理 1 和 2 可知,

$$\lim_{k \rightarrow \infty} \|e_{k+1}^{(n-1),*}(t)\|_\lambda = 0, \quad (14)$$

也就是说, 在  $[0, 2^{n-1}t_p]$  上  $\|e_{k+1}^{(n-1),*}(t)\|$  一致收敛于 0.

根据  $\Theta_i(t)$  和  $e_{k+1}^{(n-1),*}(t)$  的定义可知, 当  $t = 2^{n-i}t_p$ ,  $i = 1, 2, \dots, n$  时, 有

$$\lim_{k \rightarrow \infty} \|e_{k+1}^{(n-1)}(t)\| = \lim_{k \rightarrow \infty} \|e_{k+1}^{(n-1),*}(t)\| = 0. \quad (15)$$

**注 2** 从上面的推导过程中可知, 在  $[0, t_p]$  上, 系统 (7) 实际运行轨迹的  $n-1$  阶导数就是  $y_{m,d}^{(n-1),*}(t)$ .

以此类推, 当  $t \in [h_{i,1}, 2^{n-1}t_p]$  并且  $1 \leq i \leq n-1$  时,

$$x_{k+1}^i(t) - x_k^i(t) = \Omega + \omega_1 - \omega_2 + \omega_3 + [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(i-1)}(t). \quad (16)$$

式中,

$$\begin{aligned} \Omega &= \underbrace{\int_{h_{i,1}}^{t_{i-1}} \cdots \int_{h_{i,1}}^{t_{n-2}}}_{n-i} \Delta x_k^n(h_{i,1}) dt_{n-1} \cdots dt_i + \cdots \\ &\quad + \underbrace{\int_{h_{i,1}}^{t_{i-1}} \int_{h_{i,1}}^{t_i} \Delta x_k^{i+2}(h_{i,1}) dt_i}_{2} + \underbrace{\int_{h_{i,1}}^{t_{i-1}} \Delta x_k^{i+1}(h_{i,1}) dt_i}_{1} \\ \omega_1 &= \int_{h_{i,1}}^{t_{i-1}} \cdots \int_{h_{i,1}}^{t_{n-1}} (\mathbf{I}_{m_a} \otimes \mathbf{A}) \Delta x_k^n(t_n) dt_n \cdots dt_i \\ \omega_2 &= \sum_{j=i}^n \frac{t^{j-i}}{(j-i)!} [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] e_k^{(j-1)}(h_{i,1}) \\ \omega_3 &= \underbrace{\int_{h_{i,1}}^t \cdots \int_{h_{i,1}}^{t_{n-1}}}_{n-i+1} \sum_{j=1}^i [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma] * \Theta_j(t_n) e_k^{(j-1)}(h_{i,1}) dt_n \cdots dt_i. \end{aligned}$$

同样假设下面的轨迹

$$y_{m,d}^{(i-1),*}(t) = y_{m,d}^{(i-1)}(t) + \omega_3^* - (\mathbf{I}_{m_a} \otimes \mathbf{I}_1) e_k^{(i-1)}(h_{i,1}) \quad (17)$$

可实现, 即当系统 (1) 存在固定初始偏差  $x_{m,d}^{(i-1),*}(0) \neq x_{m,d}^{(i-1)}(0)$  时, 存在控制量  $u_d^*(t)$  使得输出为  $y_{m,d}^{(i-1),*}(t)$ , 相应的误差记为  $e_k^{(i-1),*}(t) = y_{m,d}^{(i-1),*}(t) - y_k^{(i-1)}(t)$ . 上式中,

$$\omega_3^* = \underbrace{\int_{h_{i,1}}^t \cdots \int_{h_{i,1}}^{t_{n-1}}}_{n-i+1} \sum_{j=1}^i (\mathbf{I}_{m_a} \otimes \mathbf{I}_1) * \Theta_j(t_n) e_k^{(j-1)}(h_{i,1}) dt_n \cdots dt_i.$$

同理可得

$$e_k^{(i-1),*}(t) - e_{k+1}^{(i-1),*}(t) = \Omega + \mathbf{C}\omega_1 + [(\mathbf{L} + \mathbf{S}) \otimes \mathbf{C}B\Gamma]e_k^{(i-1),*}(t).$$

对上式进行整理后, 取  $\lambda$  范数, 如果  $\lambda$  足够大, 根据引理 1 和 2 可知,

$$\lim_{k \rightarrow \infty} \|e_{k+1}^{(i-1),*}(t)\|_\lambda = 0.$$

相应地可以得到

$$\lim_{k \rightarrow \infty} \|e_k^{(i-1)}(t)\| = 0, \quad t = h_{j,2}, \quad j = i, i-1, \dots, 1. \quad (18)$$

**注 3** 在 (18) 式的推导过程中, 首先得到的是  $\lim_{k \rightarrow \infty} \|e_k^{(n-2)}(t)\| = 0$ ,  $t = h_{j,1}$ ,  $j = n-2, n-3, \dots, 1$ , 并且在此过程需要用到  $\lim_{k \rightarrow \infty} \|e_k^{(n-1)}(t)\| = 0$ ,  $t = h_{j,1}$ ,  $j = n-1, n-2, \dots, 1$  这个结论, 以此类推, 最后得到  $\lim_{k \rightarrow \infty} \|e_k(t)\| = 0$ ,  $t = h_{1,2} = 2^{n-1}t_p$ . 从上面的推导过程中可知, 在  $[h_{i,1}, h_{i,2}]$  上, 系统 (7) 实际运行轨迹的  $i-1$  阶导数就是  $y_{m,d}^{(i-1),*}(t)$ .

根据上面的推导结果可知, 当  $t = 2^{n-1}t_p$  并且  $k \rightarrow \infty$  时, 有  $x_k^i(t) = x_{m,d}^{(i-1)}(t)$ ,  $i = n, n-1, \dots, 1$  成立, 也就是说, 此时的  $x_k^{(i-1)}(t)$  处于理想状态.

当  $t \in [2^{n-1}t_p, T]$  时, 对于任意的  $x_{k+1}^i(t) - x_k^i(t)$ ,  $i = n, n-1, \dots, 1$ , 有

$$x_{k+1}^i(t) - x_k^i(t) = \omega_1^* + (\mathbf{L} + \mathbf{S}) \otimes \mathbf{B}\Gamma e_k^{(i-1)}(t), \quad (19)$$

其中,  $\omega_1^* = \int_{h_{1,2}}^t \cdots \int_{h_{1,2}}^{t_{n-1}} (\mathbf{I}_{m_a} \otimes \mathbf{A}) \Delta x_k^n(t_n) dt_n \cdots dt_i$ .

在 (19) 式的两端同时乘以矩阵  $\mathbf{C}$  并取  $\lambda$  范数, 很容易得到

$$\lim_{k \rightarrow \infty} \|e_k^{(i-1)}(t)\|_\lambda = 0.$$

证毕.

**注 4** 初始修正的过程可以描述如下: 系统 (1) 根据控制算法 (3) 首先修正  $x_k^n$ , 然后  $x_k^{n-1}, x_k^{n-1}, \dots, x_k^2$ , 最后在  $t = 2^{n-1}t_p$  时刻完成修正  $x_k^1$ . 当修正状态  $x_k^i$  时, 状态  $x_k^j$  ( $j = i+1, i+2, \dots, n$ ) 已经修正好, 并且状态  $x_k^i$  将在  $t = 2^{n-i}t_p$  时刻被修正好. 从函数  $\theta_i$  的定义可知: 当  $i < n$  时, 修正时间为  $[2^{n-i-1}t_p, 2^{n-i}t_p)$  (当  $i=n$  时, 修正时间为  $[0, t_p)$ ), 因为在其他区间修正函数的函数值为零. 即使对  $x_k^i$  的修正操作已经完成, 并不意味着  $x_k^i$  在区间  $[2^{n-j-1}t_p, 2^{n-j}t_p)$  ( $j = i-1, \dots, 1$ ) 上完全跟踪上  $x_d^i$ . 关于这个观点, 可以这样解释: 因为在修正  $x_k^i$  时, 修正操作势必会影响  $x_k^j$  ( $j > i$ ). 而一旦对  $x_k^i$  的修正操作完成, 修正操作对  $x_k^j$  ( $j > i$ ) 的影响也就结束, 并且此时  $x_k^j$  ( $j > i$ ) 重新跟踪上  $x_d^j$ . 以此类推, 在  $t = 2^{n-1}t_p$  时, 系统完成所有修正操作, 并完全跟踪上期望状态  $x_d^1, \dots, x_{n,d}$ . 从某种意义上说, 这种方法的本质就是给每个状态  $x_k^i$  施加了一个脉冲, 使初始状态回到期望位置, 并且施加脉冲时还有一个先后顺序, 即首先是  $x_k^n$ , 其次是  $x_k^{n-1}, \dots, x_k^2$ , 最后是  $x_k^1$ .

**注 5** 由于在修正状态  $x_k^i$  时, 修正操作对状态  $x_k^j$  ( $j = i + 1, \dots, n - 1$ ) 的影响不可避免, 考虑到控制量的大小问题, 进行修正时建议采用如下方式: 对于状态  $x_k^i$  而言,  $i$  越小, 修正时间应越长. 因为  $i$  越小, 影响的状态  $x_k^j$  ( $j > i$ ) 越多. 比如, 修正  $x_k^1$  时, 修正操作势必会影响  $x_k^2, x_k^3, \dots, x_k^n$ .

**注 6** 本文中建议的修正函数并不是唯一的, 只要其满足定理 1 即可, 因此也可以采用下列形式.

$$\Theta_i(t) = \begin{cases} 0, & t \in [0, h_{i,1}), \\ \frac{1}{\int_{h_{i,1}}^{h_{i,2}} \theta_i(\tau) d\tau} \Theta_i(t)^{(n-i)}, & t \in [h_{i,1}, h_{i,2}), \\ 0, & t \in [h_{i,2}, T]. \end{cases}$$

式中,  $\theta_i = t^{N-i}(t - h_{i,1})^{N-i}(h_{i,2} - t)^{N-i}$ .

## 5 数值仿真

考虑在区间  $[0, T]$  重复运行的多智能体系统包含一个虚拟领导者, 四个智能体, 领导者的状态即为所要跟踪的状态, 该多智能体的拓扑图见图 1. 图中, leader 是虚拟领导者, 其他为智能体. 本节将通过仿真来验证算法的有效性, 在仿真过程中, 假设所有智能体的初始输入均为零, 也就是说, 对于  $\forall j, \forall t$ , 都有  $u_{j,0}(t) = 0$ .

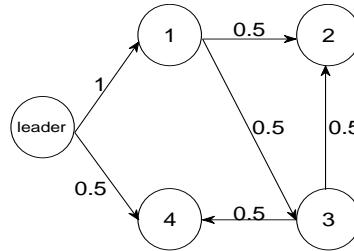


图 1 多智能体系统拓补图

图 1 中各个智能体的动态特性为

$$\begin{cases} \dot{x}_j^1(t) = x_j^2(t), \\ \dot{x}_j^2(t) = -x_j^2(t) + u(t), \\ y(t) = x_j^1(t). \end{cases}$$

多智能体系统的作业区间为  $[0, 2]$ , 状态修正区间为  $[0, 0.4]$ . 各智能体的跟踪轨迹为  $y_d(t) = x_{1,d}(t) = \cos(\pi t)$ ,  $x_{2,d}(t) = -\pi \sin(\pi t)$ ,  $x_{3,d}(t) = -\pi^2 \cos(\pi t)$ ,  $t \in [0, 2]$ . 系统运行之前初始状态为  $x_{1,k}^1(0) = 1.5$ ,  $x_{1,k}^2(0) = -1$ ;  $x_{2,k}^1(0) = 0.8$ ,  $x_{2,k}^2(0) = -3$ ;

$x_{3,k}^1(0) = 0.1, x_{3,k}^2(0) = 2; x_{4,k}^1(0) = -0.5, x_{4,k}^2(0) = 5$ , 以下各图都是迭代次数为 20 次时的仿真结果.

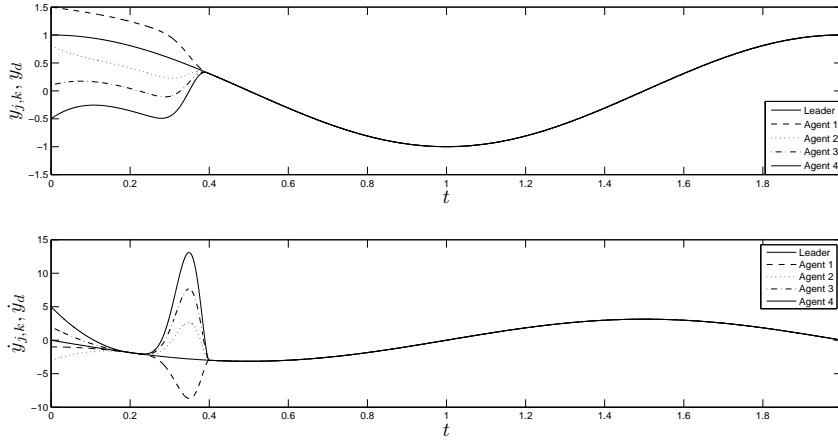


图 2 领导者的  $y_d(t)$  和  $\dot{y}_d(t)$  与各智能体的  $y_{j,k}(t)$  和  $\dot{y}_{j,k}(t)$

从图 2 中可以看出控制律在区间  $t = 0.2$  时完成修正状态  $\dot{y}_k(t)$  的误差, 但此时并没有修正  $y_k^1(t)$  的跟踪误差, 因此各智能体的实际轨迹  $y_k(t)$  并没有跟踪上领导者的轨迹  $y_d^1(t)$ . 实际上,  $y_k(t)$  的跟踪误差将在区间  $[0.2, 0.4]$  上被修正. 而且, 在修正  $y_k^1(t)$  的跟踪误差期间,  $y_k^2(t)$  也没能跟踪上  $y_d^1(t)$ . 并且在修正  $y_k^1(t)$  时, 实际状态  $y_k^2(t)$  不仅变化幅度比较大而且变化比较快. 这很容易理解, 实际上  $x_k^2(t)$  代表速度, 修正  $x_k^1(t)$  也就是改变位移, 当花费时间很少时必然要求速度很大. 当  $t = 0.4s$  时, 修正  $x_k^1(t)$  操作完成, 系统完全跟踪上领导者轨迹.

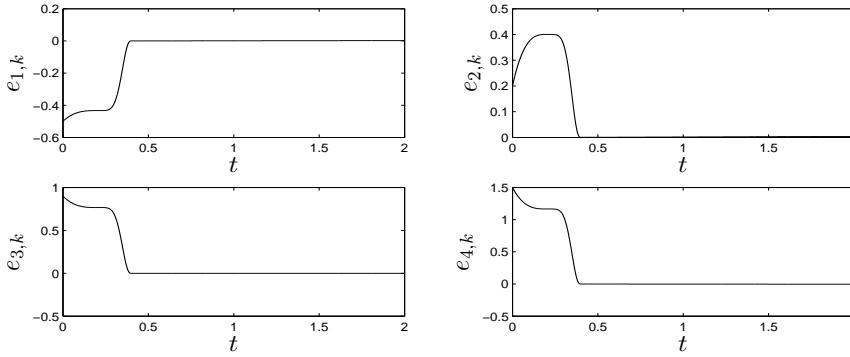


图 3 各智能体的实际跟踪误差  $e_{j,k}^1(t)$

图 3 和图 4 中,  $\dot{e}_{1,k}(t)$ ,  $\dot{e}_{2,k}(t)$ ,  $\dot{e}_{3,k}(t)$  和  $\dot{e}_{4,k}(t)$  分别代表智能体 1 至 4 的状态误差,  $e_{1,k}(t)$ ,  $e_{2,k}(t)$ ,  $e_{3,k}(t)$  和  $e_{4,k}(t)$  分别代表智能体 1 至 4 的跟踪误差.

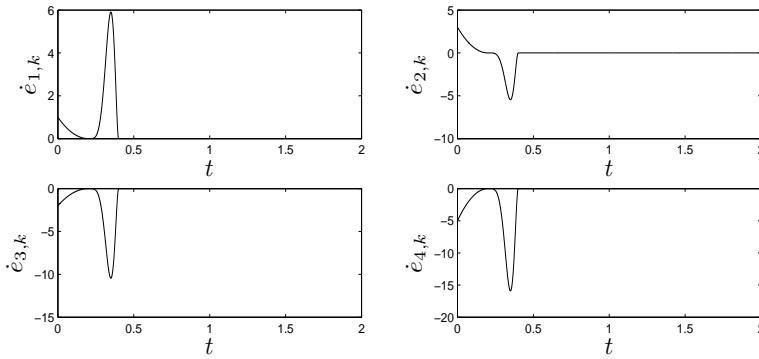


图 4 各智能体的实际状态误差  $e_{j,k}^2(t)$

从图 5 可以看出, 在修正状态  $x_k^1(t)$  时, 需要很大的控制量. 考虑到控制量的限制, 一般要求修正时间不能太短.

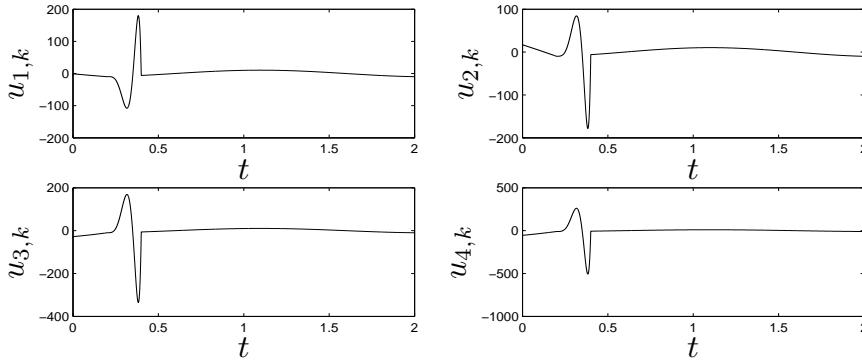


图 5 各智能体的控制量  $u_{j,k}(t)$

上面的仿真验证了本文所提算法对于多智能体系统而言, 不仅能够完成初始状态误差修正, 而且能够达到一致性跟踪.

## 6 结论

本文研究了高阶线性多智能体的一致性跟踪问题, 提出了一种 step-by-step 修正方法, 应用此方法时允许所有智能体的各阶状态都带有状态误差. 在跟踪过程中, 该控制算法不需要通过初态学习就能在指定区间达到一致性跟踪. 在实现时该算法首先修正每一个智能体的最高阶状态误差即  $x_{j,k}^n(t)$  的误差, 然后是  $x_{j,k}^i(t)$ ,  $i = n-1, \dots, 2$ , 最

后修正  $x_{j,k}^1(t)$  的状态误差, 而且所有的修正操作在一个指定时间内完成. 当修正好所有的状态误差后, 多智能体系统即跟踪上期望轨迹, 而且本文从理论和实践两方面验证了此算法的有效性, 但本文方法不适用于线性时变多智能体系统和异质多智能体系统, 这也是今后研究和努力的方向.

## 参 考 文 献

- [1] Vicsek T, Czirók A, Ben-Jacob E, Cohen I, Shochet O. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 1995, 75(6): 1226–1229
- [2] Jadbabaie A, Lin J, Morse A. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control*, 2003, 48(6): 988–1001
- [3] 关永强, 纪志坚, 张霖, 王龙. 多智能体系统能控性研究进展. *控制理论与应用*, 2015, 32(4): 421–431  
(Guan Y, Ji Z, Zhang L, Wang L. Recent developments on controllability of multi-agent systems. *Control Theory & Applications*, 2017, 34(3): 401–407)
- [4] Ren W, Beard R, Atkins E. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 2007, 27(2): 71–82
- [5] Meng D, Jia Y. Formation control for multi-agent systems through an iterative learning design approach. *International Journal of Robust & Nonlinear Control*, 2014, (24): 340–361
- [6] Fax J, Murray R. Information flow and cooperative control of vehicle formations. *IEEE Trans. Autom. Control*, 2004, 49(9): 1465–1476
- [7] Olfati-Saber R, Murray M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control*, 2004, 49(9): 1520–1533
- [8] Zhu W, Cheng D. Leader-following consensus of second-order agents with multiple time-varying delays. *Automatica*, 2010, 46(12): 1994–1999
- [9] Ren W. Second-order consensus algorithm with extensions to switching topologies and reference models. In: Proceedings of the IEEE American Control Conference. New York: IEEE, 2007, 1431–1436
- [10] Khan U. High Dimensional Consensus in Large-scale Networks: Theory and Applications. Carnegie Mellon University, USA, 2009
- [11] Huang Q. Consensus analysis of multi-agent discrete-time systems. *Acta Automatica Sinica*, 2012, 38(7): 1227–1233
- [12] 陈世明, 王培, 李海英, 赖强. 带强连通分支的多智能体系统可控包含控制. *控制理论与应用*, 2017, 34(3): 401–407  
(Chen S, Wang P, Li H, Lai Q. Controllable containment control of multi-agent systems with strongly connected sub-graph. *Control Theory & Applications*, 2017, 34(3): 401–407)
- [13] 盖彦荣, 陈阳舟, 宋学君, 齐耀辉. 有领导者线性多智能体系统一致性的分析与设计. *中南大学学报(自然科学版)*, 2017, 48(3): 735–741  
(Ge Y, Chen Y, Song X, Qi Y. Consensus analysis and design problem for leader-following linear

- multi-agent systems. *Journal of Central South University (Science and Technology)*, 2017, 48(3): 735–741)
- [14] 谭拂晓, 关新平, 刘德荣. 非平衡拓扑结构的多智能体网络系统一致性协议. *控制理论与应用*, 2009, 26(10): 1087–1092  
(Tan F, Guan X, Liu D. Consensus protocol in networked multi-agent systems with non-balanced topology. *Control Theory & Applications*, 2009, 26(10): 1087–1092)
- [15] 席裕庚, 李晓丽. 多智能体系统一致性的递阶结构设计. *控制理论与应用*, 2015, 32(9): 1191–1199  
(Xi Y, Li X. Hierarchical structure design for multi-agent consensus. *Control Theory & Applications*, 2015, 32(9): 1191–1199)
- [16] Das A, Lewis F. Distributed adaptive control for synchronization of unknown nonlinear networked systems. *Automatica*, 2010, 46(12): 2014–2021
- [17] Ranjbar S, Shabaninia F, Nemati A, Stan S. A novel robust decentralized adaptive fuzzy control for swarm formation of multiagent systems. *IEEE Transactions on Industrial Electronics*, 2013, 59(8): 3124–3134
- [18] Arimoto S, Kawamura S, Miyazaki F. Bettering operation of robots by learning. *J. Robotic Syst.*, 1984, 1(2): 123–140
- [19] Bristow D, Tharayil M, Alleyne A. A survey of iterative learning control. *IEEE Control Syst. Mag.*, 2006, 26(3): 96–114
- [20] Meng D, Jia Y. Finite-time consensus for multi-agent systems via terminal feedback iterative learning. *IET Control Theory and Applications*, 2011, 5(18): 2098–2110
- [21] Meng D, Jia Y. Iterative learning approaches to design finitetime consensus protocols for multi-agent systems. *Systems and Control Letters*, 2012, 61(1): 187–194
- [22] Meng D, Jia Y, Du J, Yu F. Tracking control over a finite interval for multi-agent systems with a time-varying reference trajectory. *Systems & Control Letters*, 2012, (67): 807–818
- [23] Wang D. Convergence and robustness of discrete time nonlinear systems with iterative learning control. *Automatica*, 1998, 34(11): 1445–1448
- [24] Sun M, Wang D. Iterative learning control with initial rectifying action. *Automatica*, 2002, 38(7): 1177–1182
- [25] Li X, Chow T, Ho J, Zhang J. Iterative learning control with initial rectifying action for nonlinear continuous systems. *IET Control Theory Appl.*, 2009, 3(1): 49–55
- [26] Xu J, Tan Y. A composite energy function-based learning control approach for nonlinear systems with time-varying parametric uncertainties. *IEEE Trans. Autom. Control*, 2002, 47(11): 1940–1945
- [27] Qu Z, Xu J. Asymptotic learning control for a class of cascaded nonlinear uncertain systems. *IEEE Trans. Autom. Control*, 2002, 47(8): 1369–1376
- [28] Chien C, Hsu C, Yao C. Fuzzy system-based adaptive iterative learning control for nonlinear plants with initial state errors. *IEEE Trans. Fuzzy Syst.*, 2004, 12(5): 724–732
- [29] Xu J, Yan R. On initial conditions in iterative learning control. *IEEE Trans. Autom. Control*, 2005, 50(9): 1349–1354

- 
- [30] Chi R, Hou Z, Xu J. Adaptive ILC for a class of discrete-time systems with iteration-varying trajectory and random initial condition. *Automatica*, 2008, 44(8): 2207–2213
  - [31] 李金沙. 多智能体系统一致性学习协议的设计与分析. 西安电子科技大学, 2015  
(Li J. Design and analysis of consensus protocols for multi-agent systems by using learning control. Xi'an: Xidian University, 2015)
  - [32] 严求真, 孙明轩, 李鹤. 非参数不确定多智能体系统一致性误差跟踪学习控制. 控制理论与应用, 2016, 33(6): 793–799  
(Yan Q, Sun M, Li H. Consensus-error-tracking learning control for nonparametric uncertain multi-agent systems. *Control Theory & Applications*, 2016, 33(6): 793–799)

## Consensus Tracking of High-order Multi-agent Systems with Initial State Errors

LI GUOJUN

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

(Basic Courses Department, Zhejiang Police College, Hangzhou 310053, China)

(E-mail: liguojun@zjjcxy.cn)

CHEN DONGJIE HAN YISHI

(Basic Courses Department, Zhejiang Police College, Hangzhou 310053, China)

(E-mail: chendongjie@zjjcxy.cn; hanishi@zjjcxy.cn)

**Abstract** In this paper, the problem of how to control the high-order multi-agent systems is considered by iterative learning method. During the process of tracking, the consistency tracking is realized after rectifying the initial state errors. In the process of rectifying, the systems only rectify a type of the state errors within a certain interval. When this type of state errors rectifying operation is completed, the systems begin to rectify another type of state errors, and so on. All the initial state errors can be completely rectified, and all the rectifying operations are completed within a specified time. Finally, the simulation results verify the effectiveness of the proposed algorithm.

**Key words** multi-agent; iterative learning control; directed graph; convergence;  
step-by-step rectifying

**MR(2000) Subject Classification** 93E35

**Chinese Library Classification** O212.7