Iterative Learning Control for Consensus Tracking of Multi-agent Systems

Deyuan Meng

The Seventh Research Division, Beihang University (BUAA), Beijing 100191, P. R. China

December 13, 2014

Research Overview

Research Interests:

- Iterative Learning Control (ILC)
  - robustness against model uncertainty
  - monotonic convergence
  - disturbance rejection
  - initial shift issue
- Distributed Control of Multi-agent Systems (DCMS)
  - consensus tracking
  - relative formation
- Combined Studies of ILC and DCMS
  - design of distributed ILC algorithm
  - convergence under different network topologies

Outline

- Motivations
- Basic ILC Framework
- Illustrative Example

Motivations

A motivating example of ILC is that

- “a basketball player shooting a free throw from a fixed position can improve his or her ability to score by practicing the shot repeatedly
  - During each shot, the basketball player observes the trajectory of the ball and consciously plans an alteration in the shooting motion for the next attempt
  - As the player continues to practice, the correct motion is learned and becomes ingrained into the muscle memory so that the shooting accuracy is iteratively improved

  ➥ This type of learned open-loop control strategy is the essence of ILC”

—— Bristow, Tharayil, & Alleyne (2006)

Bristow D.A., Tharayil M., and Alleyne A.G. (2006). A Survey of Iterative Learning Control: A Learning-based Method for High-performance Tracking Control. *IEEE Control Systems Magazine*, 26(3): 96–114

Motivations

Summarizing, we can see the key feature of ILC as

- "Iterative learning control (ILC),
  - a technique that attempts to refine the performance of systems that repeat their operation over and over from the same initial conditions,
  - is fundamentally a two-dimensional process, with evolution along both a finite time axis (denoted by t) and an infinite iteration axis (denoted by k)"

—— Moore, Ahn, & Chen (2008)

Moore K.L., Ahn H.S., and Chen Y.Q. (2008). Iteration Domain H_∞-optimal Iterative Learning Controller Design. *International Journal of Robust and Nonlinear Control*, 18(10): 1001–1017

- Application areas for ILC include:
  - Industrial robots; Hard disc drives; Batch/factory/chemical processes; etc

Basic ILC Framework

As is observed by the previous statement of Moore, Ahn, & Chen (2008), ILC creates a 2-D process, where

- it has two independent axes: a finite time axis and an infinite iteration axis , e.g., discrete-time ILC has
  - $t \in \mathbb{Z}_N = \{0, 1, \cdots, N\}$ — the finite time step
  - $k \in \mathbb{Z}_+ = \{0, 1, 2, \cdots\}$ — the infinite iteration number
- the control system is repetitive , e.g., a repetitive single-input single-output (SISO) system given by

$$\begin{cases} x_k(t+1) = Ax_k(t) + bu_k(t) + w_k(t) \\ y_k(t) = cx_k(t) + v_k(t), \qquad \forall x_k(0) \end{cases} \tag{1}$$

where $x_k(t) \in \mathbb{R}^{n_s}$ is the state, $u_k(t) \in \mathbb{R}$ the control input, $y_k(t) \in \mathbb{R}$ the output, and

- the system matrices A, b, c can be time-dependent but must be iteration-invariant (repetitive)
- the initial state $x_k(0)$ and external disturbances $w_k(t)$ and $v_k(t)$ can be iteration-dependent (nonrepetitive)

Basic ILC Framework

Objective and algorithm :

- The objective of ILC is to refine the performance of systems such that their output can achieve the perfect tracking of any prescribed desired reference trajectory

➥ For the system (1) satisfying $cb \neq 0$, the perfect tracking objective is given by

$$\lim_{k\to\infty} y_k(t) = y_d(t) \quad \text{for all } t = 1, 2, \cdots, N \tag{2}$$

where $y_d(t)$ is the desired output over $t \in \mathbb{Z}_N$

- The algorithm of ILC uses an updating law to improve the control input from one iteration to the next based on the tracking error

➥ For the system (1) with the tracking objective (2), we can apply

$$u_{k+1}(t) = u_k(t) + \gamma e_k(t+1) \tag{3}$$

where $e_k(t) = y_d(t) - y_k(t)$ is the tracking error, and γ is the learning gain to be designed

Convergence analysis :

- contraction mapping based approach
- super-vector based approach
- 2-D systems based approach
- H_∞ based approach
- ...

➥ For the system (1), let the ILC updating law (3) be applied with

$$|1 - cb\gamma| < 1 \tag{4}$$

- If the initial state and external disturbances are bounded but not convergent along the iteration axis, then the tracking error is bounded but there exists residual tracking error
- If the initial state and external disturbances are both bounded and convergent along the iteration axis, then the tracking error is bounded and convergent to zero with increasing iteration

An illustrative example (*Example 1*) :

- The repetitive system (1) is simulated with matrices as

$$A = \begin{bmatrix} 0.72 & 0.0 & 0.0 \\ 1.0 & -1.04 & -0.81 \\ 0.0 & 0.81 & 0.0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$c = \begin{bmatrix} 1.0 & -0.98 & -1.09 \end{bmatrix}$$

- The ILC updating law (3) is applied
  - using the learning gain as $\gamma = 0.6$, which guarantees

$$|1 - cb\gamma| = 0.4 < 1$$

    i.e., the convergence condition (4) holds
  - using the zero initial input, i.e., $u_0(t) \equiv 0$ for all t
- The desired reference $y_d(t)$ is chosen as

$$y_d(t) = 1.5\sin(0.02\pi t), \quad \text{for all } t \in \mathbb{Z}_{100}$$

Example 1

- The initial state and external disturbances take the form of

$$
\begin{aligned}
x_k(0) &= x_0 + \alpha(k)\varpi_x(k) \\
w_k(t) &= \alpha(k)\varpi_w(k,t) \\
v_k(t) &= \alpha(k)\varpi_v(k,t)
\end{aligned}
$$

where $x_0 = \begin{bmatrix} 1 & -3 & 4 \end{bmatrix}^{\mathrm{T}}$, and

- $\alpha(k)$ is a scalar function to be simulated in two cases as
  - $\alpha(k)$ is bounded but not convergent, where

    $$\alpha(k) = 0.1\cos(k)$$

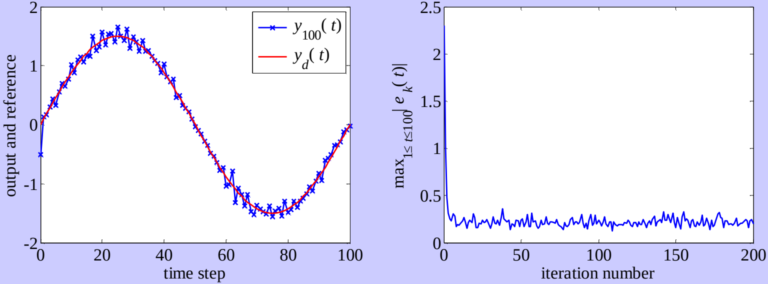
  - $\alpha(k)$ is both bounded and convergent, where

    $$\alpha(k) = 0.3(k+1)^{-1}$$

- $\varpi_x(k)$, $\varpi_w(k,t)$, and $\varpi_v(k,t)$ denote appropriately dimensioned uncertainties, and every element of them varies arbitrarily over the interval $[-0.5, 0.5]$ as a function of both iteration number k and time step t

Example 1

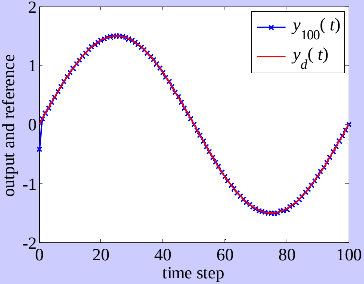
- If $\alpha(k) = 0.1\cos(k)$, i.e., the initial state and external disturbances are bounded but not convergent along the iteration axis, then the tracking performance of ILC is given by

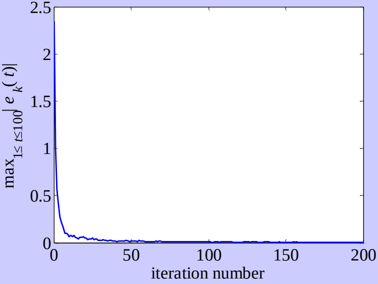


- The left figure shows that the output after $k = 100$ iterations varies in a small bound of the desired reference for all $1 \le t \le 100$
- The right figure shows that the tracking error is bounded but there exists residual tracking error

Example 1

- If $\alpha(k) = 0.3(k+1)^{-1}$, i.e., the initial state and external disturbances are both bounded and convergent along the iteration axis, then the tracking performance of ILC is given by





- The left figure shows that the output after $k = 100$ iterations can accurately follow the desired reference for all $1 \le t \le 100$
- The right figure shows that the tracking error is bounded and convergent to zero with increasing iteration

Outline

- Motivations
- Multi-agent Consensus under 2-D Switching Topologies
  - 2-D Switching Topologies
  - Problem Statement
  - ILC-Based Distributed Algorithm
  - Consensus Analysis
- Illustrative Example

Motivations

A motivating example for combined studies of ILC and DCMS is

- a group of performers that "march", or walk, through several formations while playing musical instruments
  - The goal of any individual performer is to "be at the right place at the right time", relative to the other performers
  - Such a group achieves its objectives through practice
  - ➥ http://www.wikihow.com/Practice-Marching-Band-Formations
- during each practice repetition, every performer observes the relative distance from his or her nearest neighbors and consciously plans an alteration in the formation motion for the next attempt
- as the group continues to practice, the correct motion can be learned and stored in the performers' memory such that the formation accuracy can be iteratively improved

➥ this actually shows a coordination learning picture of groups of performers whose control strategies are generated through repetition and nearest neighbor communications

Application areas for combined studies of ILC and DCMS include:

- Formation keeping of flying satellites:
  - Wu B., Poh E.K., Wang D., and Xu G. (2009). Satellite Formation Keeping Via Real-time Optimal Control and Iterative Learning Control. *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, USA, pp. 1–8
  - Ahn H.S., Moore K.L., and Chen Y.Q. (2010). Trajectory-keeping in Satellite Formation Flying via Robust Periodic Learning Control. *International Journal of Robust and Nonlinear Control*, 20(14): 1655–1666
- Formation control of mobile robots:
  - Chen X. and Jia Y. (2010). Stereo Vision-based Formation Control of Mobile Robots Using Iterative Learning. *Proceedings of the International Conference on Humanized Systems*, Kyoto, Japan, pp. 62–67

- Trajectory tracking of multiple trains
  - Sun H., Hou Z., and Li D. (2013). Coordinated Iterative Learning Control Schemes for Train Trajectory Tracking with Overspeed Protection. *IEEE Transactions on Automation Science and Engineering*, 10(2): 323–333

- Other possible applications include:
  - Synchronized swimming
  - Soldiers marching in formation
  - Aerial flight displays involving multiple aircrafts
  - ...

Motivations

Existing theoretical studies:

- The first publication of combined studies on ILC and multi-agent formation control is Ahn and Chen (2009):

  Ahn H.S. and Chen Y.Q. (2009). Iterative Learning Control for Multi-agent Formation. *Proceedings of the ICROS-SICE International Joint Conference*, Fukuoka, Japan, pp. 3111–3116

  ➥ The ring and tree-type directed graphs are considered to represent the communication between agents

- An extension of Ahn and Chen (2009) is Liu and Jia (2012):

  Liu Y. and Jia Y. (2012). An Iterative Learning Approach to Formation Control of Multi-agent Systems. *Systems and Control Letters*, 61(1): 148–154

  ➥ The general directed graph is considered to represent the communication between agents

- Others: Xu et al. (2011, 2013, 2014); Li and Li (2013, 2014); Shi, He, Wang, and Zhou (2014); etc

2-D Switching Topologies

Next, we discuss combined studies of ILC and consensus for multi-agent systems with n agents under 2-D switching topologies that change along both a finite time axis and an infinite iteration axis, where

- each agent is regarded as a vertex of a dynamically changing communication nth order directed graph $\mathcal{G}_k(t)$, for all $t \in \mathbb{Z}_N$ and $k \in \mathbb{Z}_+$
- each edge $(v_i, v_j) \in \mathcal{E}(\mathcal{G}_k(t))$ denotes an available information channel from the agent v_j to the agent v_i at time step t and iteration k
- the weighted adjacency matrix associated with $\mathcal{G}_k(t)$ is denoted by $\mathcal{A}_k(t) = \left[a_{ij,k}(t)\right]$
- the Laplacian matrix associated with $\mathcal{A}_k(t)$ is denoted by $\mathcal{L}_{\mathcal{A}_k(t)}$
- the neighbor index set for the agent v_i associated with $\mathcal{A}_k(t)$ is denoted by $\mathcal{N}_{i,k}(t)$
- assume that $\overline{\mathcal{G}}_\sigma = \{\mathcal{G}_{\sigma_1}, \mathcal{G}_{\sigma_2}, \cdots, \mathcal{G}_{\sigma_\mu}\}$ denotes the set of all possible directed graphs defined for the agents, i.e., $\mathcal{G}_k(t) \in \overline{\mathcal{G}}_\sigma$ for all $t \in \mathbb{Z}_N$ and $k \in \mathbb{Z}_+$

Problem Statement

Model plant of each agent:

- The agent v_i has the following dynamics over $t \in \mathbb{Z}_N$ and $k \in \mathbb{Z}_+$:

$$\begin{aligned} x_{i,k}(t+1) &= Ax_{i,k}(t) + bu_{i,k}(t) + w_{i,k}(t) \\ y_{i,k}(t) &= cx_{i,k}(t) + v_{i,k}(t), \quad \forall x_{i,k}(0) \end{aligned} \quad \text{for } i \in \mathcal{I}_n \tag{5}$$

where $x_{i,k}(t)$ is the state, $u_{i,k}(t)$ is the input, $y_{i,k}(t)$ is the output, $w_{i,k}(t)$ is the state disturbance, $v_{i,k}(t)$ is the output disturbance, and A, b and c are appropriately dimensioned matrices

Problem for combined studies on ILC and multi-agent consensus:

- Given any desired reference trajectory $y_d(t)$ for $t \in \mathbb{Z}_N$, let $\Omega_k(t) = \text{diag}\{\omega_{1,k}(t), \cdots, \omega_{n,k}(t)\}$ be associated with the accessibility of $y_d(t)$ by agents in $\mathcal{G}_k(t)$, i.e., $\omega_{i,k}(t) > 0$ if $y_d(t)$ is accessible by the agent v_i at the time step t and the iteration k, and $\omega_{i,k}(t) = 0$ otherwise

➥ For $i \in \mathcal{I}_n$, the agent v_i is said to achieve the consensus tracking of the desired reference trajectory $y_d(t)$ if

$$\lim_{l \to \infty} y_{i,k}(t) = y_d(t) \quad \text{for } t = 1, 2, \cdots, N \tag{6}$$

ILC-Based Distributed Algorithm

ILC-based distributed algorithm for multi-agent consensus systems :

- Using the nearest neighbor rule, we present

$$u_{i,k+1}(t) = u_{i,k}(t) + \gamma_k(t)\Bigg\{ \sum_{j\in\mathcal{N}_{i,k}(t)} a_{ij,k}(t)\left[y_{j,k}(t+1) - y_{i,k}(t+1)\right] + \omega_{i,k}(t)\left[y_d(t+1) - y_{i,k}(t+1)\right]\Bigg\} \tag{7}$$

where $\gamma_k(t)$ is a unified learning gain to be designed for all agents

Note: If $e_{i,k}(t) = y_d(t) - y_{i,k}(t)$ is the tracking error of v_i, (7) becomes

$$u_{i,k+1}(t) = u_{i,k}(t) + \gamma_k(t)\Bigg\{ \sum_{j\in\mathcal{N}_{i,k}(t)} a_{ij,k}(t)\left[e_{i,k}(t+1) - e_{j,k}(t+1)\right] + \omega_{i,k}(t)e_{i,k}(t+1)\Bigg\}$$

➥ i.e., like classical ILC, (7) can be given in an iterative manner based on "the tracking errors $\{e_{i,k}(t) : i \in \mathcal{I}_n\}$ for all agents"

Consensus Analysis

For the multi-agent system (5) under the ILC-based distributed algorithm (7), let $x_k(t) = \left[x_{1,k}^{\mathrm{T}}(t), x_{2,k}^{\mathrm{T}}(t), \cdots, x_{n,k}^{\mathrm{T}}(t)\right]^{\mathrm{T}}$ and $\Delta x_k(t) = x_{k+1}(t) - x_k(t)$, and let $e_k(t)$, $w_k(t)$, $\Delta w_k(t)$, $v_k(t)$, and $\Delta v_k(t)$ be defined in the same way. Then

- Along the iteration axis k, the tracking error $e_k(t)$ satisfies

$$e_{k+1}(t+1) = \underbrace{\left\{I - cb\gamma_k(t)\left[\mathcal{L}_{\mathcal{A}_k(t)} + \Omega_k(t)\right]\right\}}_{iteration-dependent} e_k(t+1) - (I \otimes cA)\Delta x_k(t) - (I \otimes c)\Delta w_k(t) - \Delta v_k(t+1)$$

- Along the time axis t, the state error $\Delta x_k(t)$ satisfies

$$\Delta x_k(t+1) = (I \otimes A)\Delta x_k(t) + \left\{\left[\mathcal{L}_{\mathcal{A}_k(t)} + \Omega_k(t)\right] \otimes b\gamma_k(t)\right\} e_k(t+1) + \Delta w_k(t)$$

Note: The ILC process of multi-agent systems with 2-D switching topologies depends on iteration-dependent (or nonrepetitive) matrices

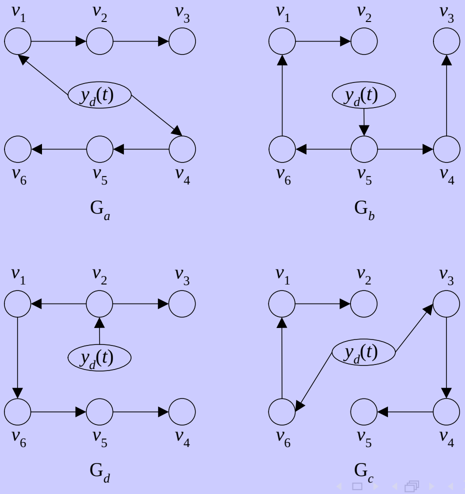
➥ The ILC convergence condition depends on $I - cb\gamma_k(t)\left[\mathcal{L}_{\mathcal{A}_k(t)} + \Omega_k(t)\right]$, which can be given by

$$\left\| I - cb\gamma_k(t)\left[\mathcal{L}_{\mathcal{A}_k(t)} + \Omega_k(t)\right] \right\| \leq \chi < 1, \quad \forall t, k$$

Example 2

An illustrative example (*Example 2*) :

- The communication graph $\mathcal{G}_k(t)$ is considered to switch over four states, i.e., $\mathcal{G}_k(t) \in \mathcal{G}_\sigma = \{\mathrm{G}_a, \mathrm{G}_b, \mathrm{G}_c, \mathrm{G}_d\}$ for all t, k, where



Example 2

- For the switching of network topologies, let $\sigma(t,k)$ be a switching signal varying on the interval [0, 1] arbitrarily as a function of both t and k, where the switching rule is detailed as
  - if $\sigma(t,k) \in [0, 0.25)$ and $\sigma(k,l) = 1$, then $\mathcal{G}_k(t) = \mathrm{G}_a$
  - if $\sigma(t,k) \in [0.25, 0.5)$, then $\mathcal{G}_k(t) = \mathrm{G}_b$
  - if $\sigma(t,k) \in [0.5, 0.75)$, then $\mathcal{G}_k(t) = \mathrm{G}_c$
  - if $\sigma(t,k) \in [0.75, 1)$, then $\mathcal{G}_k(t) = \mathrm{G}_d$
- The weight of each edge in the four directed graphs G_a, G_b, G_c, and G_d is adopted as 0.5
- The multi-agent system (5) is considered with matrices given by

$$A = \begin{bmatrix} 0.9124 & 0.0755 \\ -1.5852 & 0.5242 \end{bmatrix},\ b = \begin{bmatrix} 0.0072 \\ 0.1294 \end{bmatrix},\ c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}^{\mathrm{T}}$$

- The ILC-based distributed algorithm (7) is applied with the learning gain $\gamma_k(t) = 6.9550$, $\forall t, k$

- The initial state and external disturbances take the form of

$$x_k(0) = \begin{bmatrix} 6 & -2 & 4 & -6 & 2 & 4 \end{bmatrix}^{\mathrm{T}} \otimes 1_2 + \alpha(k)\varpi_x(k)$$
$$w_k(t) = \alpha(k)\varpi_w(k,t), \quad v_k(t) = \alpha(k)\varpi_v(k,t)$$

  where

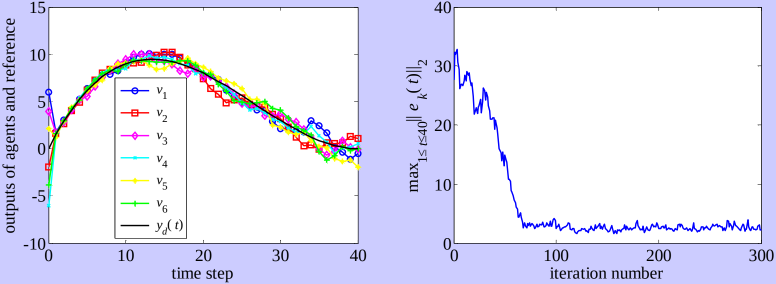
  - $\alpha(k)$ is a scalar function to be simulated in two cases as
    - $\alpha(k)$ is bounded but not convergent. Here, $\alpha(k) = 1, \forall k$
    - $\alpha(k)$ is bounded and convergent. Here, $\alpha(k) = (k+1)^{-1}, \forall k$
  - $\varpi_x(k)$, $\varpi_w(k,t)$, and $\varpi_v(k,t)$ denote appropriately dimensioned uncertainties, and every element of them varies arbitrarily over the interval $[-0.1, 0.1]$ as a function of both iteration number k and time step t

- We perform simulation with the zero initial input, i.e., $u_0(t) = 0$ for all t, and the following desired reference $y_d(t)$:

$$y_d(t) = (0.1t)^3 - 8(0.1t)^2 + 1.6t, \quad \text{for all } t \in \mathbb{Z}_{40}$$

Example 2

If the initial state and external disturbances are bounded but not convergent, then the consensus tracking performance of agents is given by

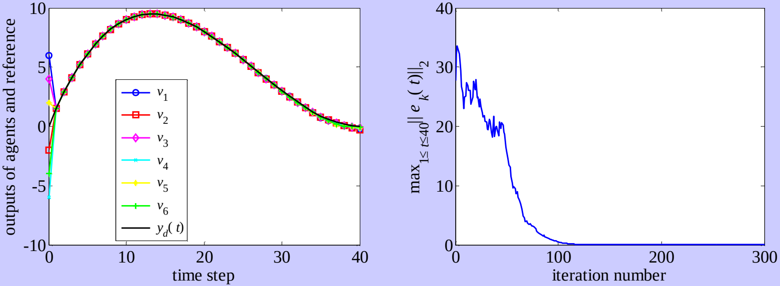


- Left: The output trajectories of agents after $k = 100$ iterations
- Right: The iterative learning process evaluated by $\max_{1\leq t\leq 40} \|e_k(t)\|_2$ for the multi-agent system over the first 300 iterations

➥ Obviously, the output trajectories of all agents vary in a small bound of the desired reference over the finite time steps $1 \leq t \leq 40$

Example 2

If the initial state and external disturbances are both bounded and convergent, then the consensus tracking performance of agents is given by



- Left: The output trajectories of agents after $k = 100$ iterations
- Right: The iterative learning process evaluated by $\max_{1\leq t\leq 40} \|e_k(t)\|_2$ for the multi-agent system over the first 300 iterations

➥ Obviously, the output trajectories of all agents can accurately follow the desired reference over the finite time steps $1 \leq t \leq 40$

Conclusions

We have reviewed ILC and its application in networked multi-agent systems, where

- we make a generic overview of ILC, including
  - basic ILC framework
  - illustrative example
- we present discussions on applying ILC to solve multi-agent consensus tracking problems, including
  - ILC-based distributed consensus algorithm
  - convergence under 2-D switching topologies
  - illustrative example

➥ we know that ILC is applicable to systems with

- a single individual plant
- multiple individual plants that interact with each other in a networked environment

References

- Li J. and Li J. (2013). Adaptive Iterative Learning Control for Consensus of Multi-agent Systems. *IET Control Theory and Applications*, 7(1): 136–142
- Li J. and Li J. (2014). Adaptive Fuzzy Iterative Learning Control with Initial-State Learning for Coordination Control of Leader-following Multi-agent Systems. *Fuzzy Sets and Systems*, 248: 122–137
- Shi J., He X., Wang Z., and Zhou D. (2014). Iterative Consensus for a Class of Second-order Multi-agent Systems. *Journal of Intelligent and Robotic Systems*, 73(1-4): 655–664
- Yang S., Xu J.X., and Yu M. (2013). An Iterative Learning Control Approach for Synchronization of Multi-agent Systems under Iteration-varying Graph. *Proceedings of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, pp. 6682–6687
- Yang S., Xu J.X., Huang D., and Tan Y. (2014). Optimal Iterative Learning Control Design for Multi-agent Systems Consensus Tracking. *Systems and Control Letters*, 69: 80–89
- Xu J.X., Zhang S., and Yang S. (2011). A HOIM-based Iterative Learning Control Scheme for Multi-agent Formation. *Proceedings of the IEEE International Symposium on Intelligent Control*, Denver, CO, USA, pp. 418–423

Acknowledgements

Thanks for Attention!