The CSI Journal on
Computer Science and Engineering
Vol. 18, No. 1, 2020
Pages 54-59
Regular Research Paper

Nature-inspired and teaching-learning-based methods for improving convergence speed in multi-agent systems

Ramin Fotouhi and Mahdi Pourgholi

Department of Control Engineering, Shahid Beheshti University, Tehran, Iran

Abstract

This paper suggests a novel method for inverse optimal control of the multi-agent systems (MAS) via a linear quadratic regulator (LQR) based on meta-heuristic algorithms. In this regard, first, the consensus protocol is designed and then the cost function is optimized via Jaya algorithm (JA), teaching-learning algorithm (TLBO), a novel meta-heuristic algorithm called advanced teaching-learning (ATLBO) and water cycle algorithm (WCA). ATLBO consists of two phases with two random values in both phases which affect the convergence rate. The optimal value of the controller's parameter is obtained via these algorithms. Simulation outputs show the usefulness of nature-inspired and learning-based methods to calculate the cost with a better convergence rate. This research consists of an inverse optimal control approach and meta-heuristic algorithms for solving the consensus problem with the least cost.

Keywords: Control systems, optimal control, multi-agent systems, algorithms, distributed systems

1. Introduction

A myriad of researchers put great emphasis on linear quadratic regulators (LQR) and multi-agent systems because of their high importance in control subjects and other purposes. [1] suggests a protocol for an optimal consensus problem using the LQR approach for a global cost function based on interactions of agents. [2] works on optimal consensus problem based on LQR approach for two global cost functions considering agents' interactions and regardless of these interactions. A protocol based on distributed LQR is suggested for consensus of MAS which consists of quadrotors and two-wheeled mobile robots [3]. An LQR approach is proposed for optimal leader-follower consensus in MAS application which includes first-order and second-order agents considering the minimum cost [4]. [5] and [6] propose meta-heuristic algorithms to improve the convergence rate in the optimal consensus of multi-agent systems based on LQR approach.

Available memory, the complexity of the problem and cost function impact on convergence rate which requires powerful optimization techniques to minimize the number of iterations. In this paper, we suggest a novel meta-heuristic algorithm called teaching-learning-based optimization algorithm (ATLBO) which improves the convergence rate in the multi-agent systems as opposed to the other well-known algorithms. Also, this research shows the usefulness of the combination of an inverse optimal control approach and meta-heuristic algorithms for solving the consensus problem of MAS considering the minimum costs. This method would be useful for power systems and renewables which affects the operation costs, efficiency and system performance.

Table 1. List of principal symbols

Symbol	Meaning
A	Adjacency Matrix
L	Laplacian Matrix
J_R	Cost Function
β	Scaling Factor
r	Stochastic Value between 0 and 1
G	Connection Graph
$\Gamma_{j.k.i}$	value of j^{th} variable during, i^{th} iteration for k^{th} candidate
Δt	Sampling Period
K	Discrete-Time Index

2. Preliminaries

2.1 Graph theory notions

The interactions among n agents in a multi-agent system can be modeled by an undirected connection graph G which consists of a set of nodes $\mathcal{V} = \{1,\ldots,n\}$, a set of edges $\varepsilon \subseteq \upsilon \times \upsilon$, and an adjacency matrix $A=[a_{ij}]\in R^{n\times n}$ in case of an undirected and weighted graph. The degree matrix (D) is a diagonal matrix that contains information about the weighted degree of each node. The Laplacian matrix is shown as follows:

$L= D - A$

$L=[l_{ij}]\in R^{n\times n}$

$$l_{ii} = \sum_{j=1}^{n} a_{ij}\ , l_{ij} = -a_{ij}\ , i \neq j \quad (1)$$

The sum of the entries of each row in the Laplacian matrix is zero which implies that L has a zero eigenvalue and the associated eigenvector is 1_n. The zero eigenvalue of the Laplacian matrix is simple if and only if the undirected graph G is connected [7]. In the case of an undirected graph, agents can reach consensus if and only if the Laplacian matrix has a simple zero eigenvalue [8]. The dynamic of the system is defined as follows:

$$x_i[k+1] = x_i[k] + \Delta t u_i[k]. \quad (2)$$

Where Δt is the sampling period and $x_i[k].u_i[k]$ denote the state and control input of the ith agent. By referring to the cost function adopted for the optimal control of linear differential equations, the proposed cost function for system (2) is defined as follows [2]:

$$J_R = \sum_{k=0}^{\infty} \sum_{i=1}^{n} \sum_{j=1}^{i-1} a_{ij}\{\, x_i[k] - x_j[k]\}^2 + \sum_{k=0}^{\infty} \sum_{i=1}^{n} u_i^2[k] \quad (3)$$

Where in the cost function a_{ij}s are elements of the adjacency matrix, k is the discrete-time index and β is the scaling factor. Since (3) depends on the weighted adjacency matrix, it is called the interplay-related cost function.

The control problem is $min_\beta\, J_R$, subject to (2) and

$$u_i[k] = -\sum_{j=1}^{n} \beta a_{ij}\{x_i[k] - x_j[k]\}. \quad (4)$$

This controller uses fixed-structure control systems. It means that the control systems have predefined architectures and controller structures. Also, the degree of freedom is equal to one because the only parameter which can vary is the scaling factor. The motivation behind the expression of the cost function (3) is to measure the control effort and the consensus error. In this regard, according to the optimal control concepts the main aim is to determine the control input to minimize the consensus error in (3). The main goal of the inverse optimal control approach is to design a global cost function as well as distributed protocol. The Laplacian matrix is inverse optimal with regard to the interplay-related cost function. It means that the inverse optimal control approach uncovers the interplay-related cost function. This approach explains well the optimal performance to reach a consensus. The optimal scaling factor is obtained for the pre-specified Laplacian matrix considering its connection graph and its interplay-related cost function.

Theorem 1: Any symmetric Laplacian matrix $L \in R^{n\times n}$ with a simple zero eigenvalue is the optimal symmetric Laplacian matrix for cost function $J = \sum_{k=0}^{\infty}\{X^T[k]Q\, X[k] + U^T[k]RU[k]\}$ [2].

In (3), a_{ij} is the weight of the edge from node i to node j in the adjacency matrix. After combining equation (2) with (4), closed-loop system is defined as follows:

$$\dot{X}[k]=-\beta LX[k],\ \ X[k]= [x_1[k], x_2[k],\ldots, x_n[k]]^T \quad (5)$$

Accordingly, the cost function (3) can be written as:

$$J_R = \sum_{k=0}^{\infty}\{X^T[k]L\, X[k] + U^T[k]U[k]\} \quad (6)$$

Equation (6) consists of two parts: one for minimizing the consensus error and the other for the control-effort problem.

For an undirected graph G all the subsequent statements are equivalent [9]:

1) The second smallest eigenvalue of the Laplacian matrix is positive.
2) The graph is linked.
3) There is one path or more between each pair of nodes

2.2 Optimal Scaling Factor via cost function

According to the previous concepts, interplay-related cost function and optimal control problem are as follows:

$$min_\beta J_R = \sum_{k=0}^{\infty}\{X^T[k]L\, X[k] + U^T[k]U[k]\}$$

Subject to: $X[k+1] = X[k] + \Delta t U[k].\, U[k] = -\beta LX[k]$

(7)

Theorem 2: For optimal control (7), where the Laplacian matrix has a simple zero eigenvalue, the optimal scaling factor β_{opt} is defined as follows [2]:

$$\beta_{opt} = \sqrt{\frac{X^T(0)X(0) - X^T(0)m_1 m_1^T X(0)}{X^T(0)LX(0)}}\ \ ,\ \ m_1 = \frac{1_n}{\sqrt{n}} \quad (8)$$

$$(-\Delta t + \sqrt{\Delta t^2 + \left(4/_{\lambda_n}\right)}\ /2) \le \beta_{opt} \le (-\Delta t + \sqrt{\Delta t^2 + \left(4/_{\lambda_2}\right)}\ /2) \quad (9)$$

3. Optimization methods

Jaya algorithm is an efficient optimization technique where the solutions of the problem always go to the best output and go away from the worst output [10]. At any iteration i, two groups of candidates are determined for an assumed cost function. In this regard, the best candidate achieves the best amount for a supposed cost function, and the worst candidate achieves the worst amount of the assumed cost function. The main advantage of this algorithm is that the worst solution approaches the best solution step by step. $\Gamma_{j.k.i}$ is the value of the j^{th} variable for the k^{th} candidate during the i^{th} iteration, and this parameter is updated according to equation (10).

$$\Gamma'_{j.k.i} = \Gamma_{j.k.i} + r_{1.j.i}(\Gamma_{j.best.i} - |\Gamma_{j.k.i}|) - r_{2.j.i}(\Gamma_{j.worst.i} - |\Gamma_{j.k.i}|) \quad (10)$$

$\Gamma_{j.best.i}$ and $\Gamma_{j.worst.i}$ are the best and worst candidate values for the variable j during the i^{th} iteration. $r_{1.j.i}$ and $r_{2.j.i}$ are random numbers between 0 and 1 for j^{th} variable during, i^{th} iteration. If $\Gamma'_{j.k.i} > \Gamma_{j.k.i}$, then the previous output is updated otherwise the previous solution is kept. WCA is a powerful method based on population, and it is inspired by the nature cycle, which is observed in wildlife and the environment. This technique is inspired by the streams rill toward the sea and rivers [12]. In nature, streams are formed in the highlands finally move toward the sea, collected water is formed from streams and rain. This algorithm is capable of solving different problems to reach an optimal solution. WCA should be implemented via several steps as follows:

1) Defining the initial parameters
2) Creating the population on a random basis, and the matrix related to that will be as follows:

$$\text{Total population} = \begin{bmatrix} Sea \\ River_1 \\ River_2 \\ . \\ . \\ Stream_{Nsr+1} \\ . \\ . \\ Stream_{Npop} \end{bmatrix}$$

$$= \begin{bmatrix} X_1^1 & \cdots & X_N^1 \\ \vdots & \ddots & \vdots \\ X_1^{N_{pop}} & \cdots & X_N^{N_{pop}} \end{bmatrix} \quad (11)$$

N_{pop} is the total number of population. Moreover, rows and columns are the numbers of population and variables of design, respectively.

3) Calculating the cost related to the stream as follows:

$$C_i = Cost_i = f(x_1^i, x_2^i, \ldots, x_N^i), \quad i = 1,2,3,\ldots, N_{pop} \quad (12)$$

4) Determining the number of streams as follows:

$$NS_n = \text{round}\{\left| \frac{\cos t_n}{\sum_{i=1}^{N_{sr}} \cos t_i} \right| \times N_{Stream}\}, \quad n = 1,2,3,\ldots, N_{sr}$$

5) Streams rill toward the sea and rivers as follows:

$$X_{Stream}(t+1) = X_{Stream}(t) + rand \times C \times (X_{Sea}(t) - X_{Stream}(t)) \quad (13)$$

6) River tills toward the bent or sea as follows:

$$X_{River}(t+1) = X_{River}(t) + rand \times C \times (X_{Sea}(t) - X_{River}(t)) \quad (14)$$

7) Finding the best position for the river in order to generate a better solution.
8) Evaporation process can occur if the condition satisfied as follows:

$$\text{if } |X_{Sea} - X^i_{River}| < d_{max}, \quad i = 1,2,3,\ldots, N_{sr} - 1 \quad (15)$$

perform raining process via (16)
end

$d_{\max}$ is a small number close to zero.

9) New locations related to the streams are calculated as follows:

$$X_{Stream}^{new} = LB + rand \times (UB - LB) \quad (16)$$

Where LB and UB are the lower and upper bounds related to the problem.

10) Updating the value of $d_{\max}$ via (17) as follows:

$$d_{max}(t+1) = d_{max}(t) - \frac{d_{\max}(t)}{Max.iteration} \quad (17)$$

11) Checking the termination condition is satisfied otherwise come back to step 5.

The other algorithm which is a powerful effective algorithm for optimization problems is TLBO. This algorithm is inspired by the general teaching-learning procedure in a class [11]. Student and teacher are the main elements of this algorithm, and at any iteration *i*, the best student of the class is appointed as a teacher. According to the teaching- learning algorithm, a novel meta-heuristic algorithm ATLBO is proposed in this paper. ATLBO is described by two phases: one for the teacher and the other one for the student, and double random values are used in the two phases. In the teacher phase, students' information boosts, and it increases the average knowledge of the class. At any iteration *i*, *Mean* defines the mean knowledge of the class, and $\Gamma_{teacher}$ mean knowledge of the teacher. Γ_{new} *and* Γ_{old} are defined as new knowledge of the student and previous knowledge of the student respectively. The new knowledge of a student in TLBO is updated according to equation (18).

$$\Gamma_{new} = \Gamma_{old} + r(\Gamma_{teacher} - (T_F)Mean) - r(\Gamma_{teacher} + (T_F)Mean) \quad (18)$$

Where r is a random value between zero and one. T_F stands for teaching factor whose value is randomly chosen, and it can be one or two. In the student phase, students share the information, and exchanging the information among them increases the students' knowledge. In this phase, Γ_{new} is the new knowledge of the student after learning from his or her companions, and Γ_{old} is the previous knowledge of the student. The new knowledge of a student in ATLBO is updated according to equation (19).

$$\Gamma_{new} = \Gamma_{old} + r(\Gamma_j - \Gamma_i) - r(\Gamma_i - \Gamma_j) \quad \text{if } \Gamma_j > \Gamma_i$$
$$\Gamma_{new} = \Gamma_{old} + r(\Gamma_i - \Gamma_j) - r(\Gamma_j - \Gamma_i) \quad \text{if } \Gamma_j < \Gamma_i \quad (19)$$

This new algorithm uses two random values in teacher and student phases while TLBO employs just a random value in two phases shown as follows:
$\Gamma_{new}=\Gamma_{old}+r(\Gamma_{teacher}-(T_F)Mean)$ used in teacher phase and $\Gamma_{new}=\Gamma_{old}+r(\Gamma_j-\Gamma_i)$ and $\Gamma_{new}=\Gamma_{old}+r(\Gamma_i-\Gamma_j)$ were used in student phase in TLBO.

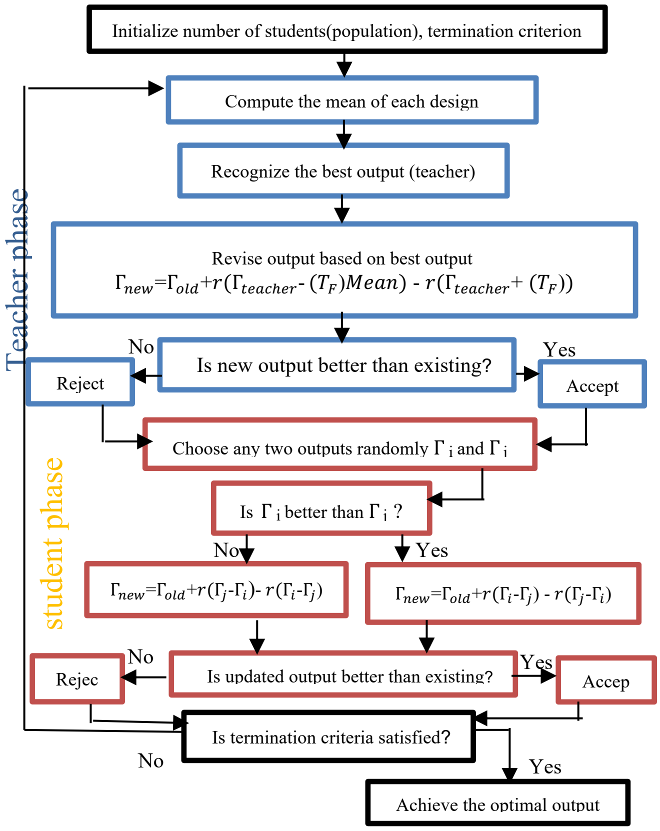

Figure 1. ATLBO flowchart

4. Consensus control for multi-robot system

The protocol introduces a new method to reach the optimal consensus. The main goal is to minimize the error between each pair of states. The calculated scaling factor obtained via ATLBO minimizes the related cost function. After several seconds all errors reach zero. This method improves the flexibility of the control system. A discrete-time controller is used, and the protocol and errors are defined as follows:

$$U(k) = -\beta LX(k) \qquad (20)$$

$$e_1 = \beta(X_2 - X_1),\ e_2 = \beta(X_3 - X_1),\ e_3 = \beta(X_1 - X_2)$$
$$e_4 = \beta(X_3 - X_2),\ e_5 = \beta(X_1 - X_3),\ e_6 = \beta(X_2 - X_3)$$
$$e_7 = \beta(X_4 - X_3),\ e_8 = \beta(X_3 - X_4) \qquad (21)$$

In order to control the agents via WCA, certain steps must be carried out as follows:

1) Designing consensus protocol
2) Optimizing the cost function via WCA
3) Finding optimal scaling factor
4) Implementing optimal scaling factor in the controller

Block diagram of the controller is as follows:

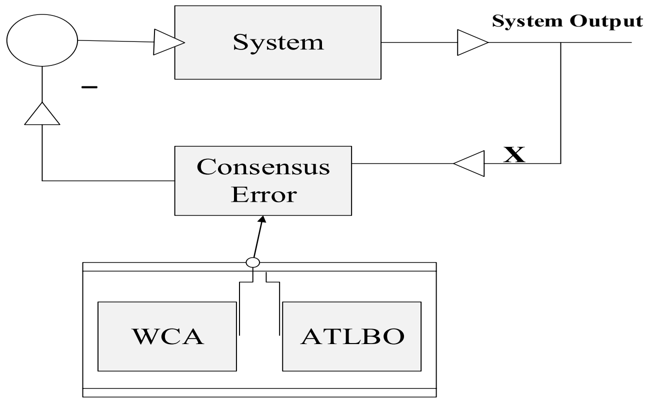

Figure 2. Block diagram of consensus controller

5. Example and results

A scenario of 4 agents is investigated to show the usefulness of the proposed technique. The communication graph among the agents is as follows:

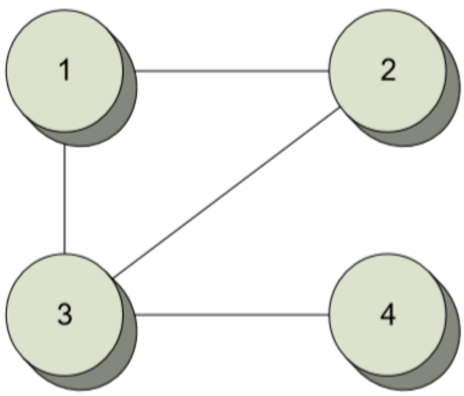

Figure 3. Connection graph of agents [2]

Laplacian matrix related to the undirected graph is the following one:

$$L=\begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \qquad (22)$$

The Laplacian matrix is symmetric, and the initial state is $X(0)=[1,2,3,4]^T$. The Laplacian matrix is inverse optimal as regards a designed cost function. In this step, the optimal value of the scaling factor is obtained using three algorithms, and this scaling factor minimizes the cost function.

Table 2. Comparison of convergence rate among algorithms

Algorithm / Iteration	JA	TLBO	WCA	ATLBO
1	107.2301	98.4215	96.3444	94.3444
2	98.1052	96.5423	95.3444	94.3444
3	94.3964	94.2319	94.9667	93.9787
4	94.2188	94.2319	93.9787	93.9787
5	94.1258	93.9991	93.8277	93.8277
6	93.9955	93.9725	93.8277	93.8271
7	93.9252	93.9148	93.8271	93.8271
8	93.9172	93.8869	93.8265	93.8265
9	93.9050	93.8523	93.8261	93.8248
10	93.8266	93.8311	93.8248	93.8248
11	93.8261	93.8311	93.8248	93.8248
12	93.8261	93.8248	93.8248	93.8248
13	93.8259	93.8248	93.8248	93.8248
14	93.8248	93.8248	93.8248	93.8248

According to Table 2, ATLBO optimizes the cost function faster than the other methods. The optimal solution for the problem is 93.8248.

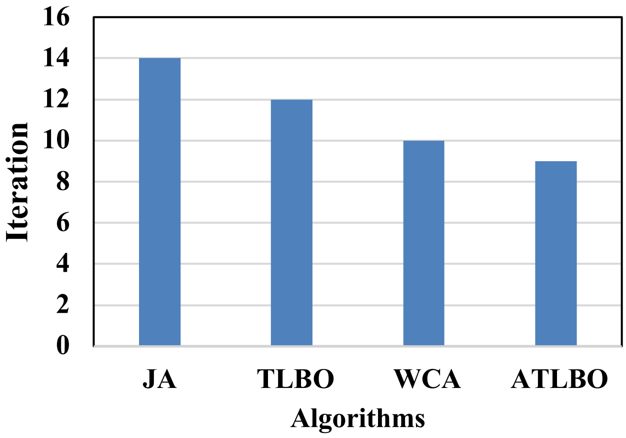

Figure 4. Comparison of the convergence rate among four methods

ATLBO minimizes the cost function after 9 iterations while JA, TLBO and WCA minimize after 14, 12 and 10 iterations respectively.

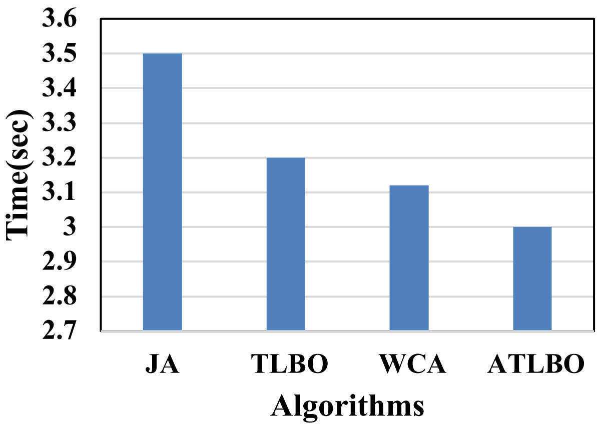

Figure 5. Runtime comparison among four methods

Comparison among all algorithms in figure 5, represents ATLBO has better output than the other methods in terms of convergence rate. The sampling period for simulations is Δt =0.1 s. The optimal value of the scaling factor is 0.471, and for this value, the cost function reaches its minimum value. The Simulation results for the states, controls and the consensus errors of the system are as follows:

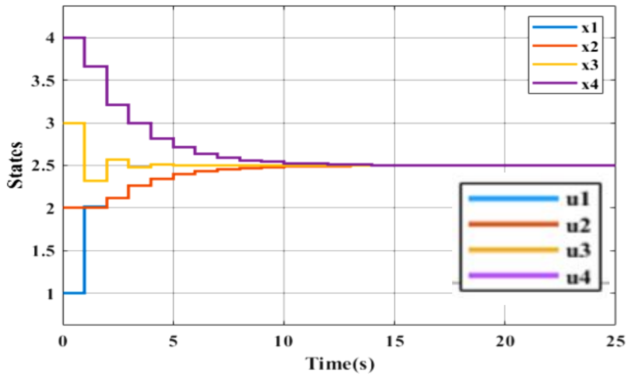

Figure 6. States of the agents

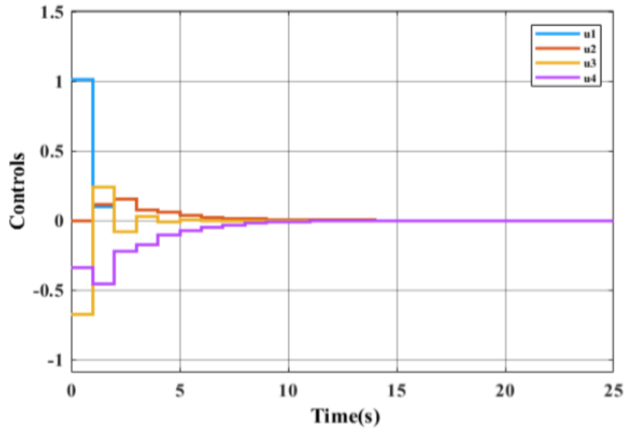

Figure 7. System control inputs

Figures 6 and 7 show the states of the agents and control inputs for agents. These simulations are obtained using ATLBO. The average consensus is obtained in figure 6.

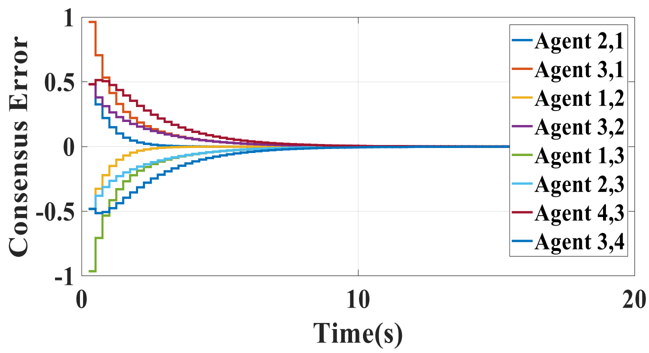

Figure 8. Consensus error of agents

In Figure 8, the eight parameters associated with the errors between each pair of agents are obtained via ATLBO. Using the regulatory problem and consensus protocol leads to an optimal consensus at zero value. The costs are calculated from zero time to the consensus time.

6. Conclusion

This paper investigates the optimal control approach and meta-heuristic algorithms to reach the consensus with the minimum cost. The global cost function is optimized via JA, TLBO, WCA and ATLBO, and the associated outputs are compared with each other. Using inverse optimal control and novel algorithms leads to an optimal consensus with improved convergence speed. The effectiveness of this novel technique for finding the optimal consensus is investigated in this research paper.

References

[1] Y. Cao and W. Ren, "LQR-based optimal linear consensus algorithms," American Control Conference, pp. 5204-5209, 2009

[2] Y. Cao and W. Ren, "Optimal Linear-Consensus Algorithms: An LQR Perspective," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 819-830, June 2010.

[3] B. Mu and Y. Shi, "Distributed LQR Consensus Control for Heterogeneous Multiagent Systems: Theory and Experiments," in *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 434-443, 2018.

[4] Y. Li, H. Yang, Y. Yang, Y. Liu, and Y. Sun, “LQR-Based Optimal Leader-Following Consensus of Heterogeneous Multi-agent Systems,” *Lecture Notes in Electrical Engineering Proceedings of 2019 Chinese Intelligent Systems Conference*, pp. 122–130, 2019.

[5] R. Fotouhi and M. Pourgholi, "Discrete-time Inverse Optimal Control for Consensus of Multi-Agent Systems via a Novel Meta-Heuristic Algorithm," *2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)*, pp. 1-5, 2021.

[6] R. Fotouhi and M. Pourgholi, "Water Cycle Algorithm-Based Control for Optimal Consensus Problem," *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, pp. 1-5, 2021.

[7] F. Chen and J. Chen, "Minimum-Energy Distributed Consensus Control of Multiagent Systems: A Network Approximation Approach," in *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1144-1159, 2020.

[8] W. Ren and R. W. Beard, “Consensus seeking in multi-agent systems under dynamically changing interaction topologies,” IEEE Transactions on Automatic Control, vol. 50, no. 5, pp. 655–661, 2005.

[9] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and Cooperation in Networked Multi-Agent Systems,” *Proceedings of the IEEE*, vol. 98, no. 7, pp. 1354–1355, 2010.

[10] R. V. Rao, “Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems,” *International Journal of Industrial Engineering Computations*, pp. 19–34, 2016.

[11] R.V. Rao, V. J. Savsani, and D. P. Vakharia, “Teaching-Learning- based optimization: A novel method for constrained mechanical design optimization problems,” Computer-Aided Design, vol. 43, no. 3, pp. 303-315, 2011.

[12] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi. "Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems." Computers & Structures, Vol. 110, pp. 151-166, 2012.

Ramin Fotouhi graduated from Urmia university, Iran, in 2016 and received his B.Sc. degree in electrical engineering. He was admitted to the Shahid Beheshti university, Iran, in 2017. He has achieved M.Sc. degree in control engineering at Shahid Beheshti university in 2019. He has published several international conference and journal papers.

Email: r.futuhi@mail.sbu.ac.ir

Mahdi Pourgholi got his B.Sc. and M.Sc. degree in 2000 and 2003 respectively. He has taken his Ph.D. degree in the electrical engineering department, Tarbiat Modares University (TMU) in 2010, Tehran, Iran. He currently is an associate professor in Shahid Beheshti University (SBU) in Tehran, Iran. He has published about 65 refereed journal and conference papers.

Email: m_pourgholi@sbu.ac.ir

Paper Handling Data:

Submitted: 04-24-2021
Received in revised form: 10-24-2021
Accepted: 11-12-2021
Corresponding author: Ramin Fotouhi
Department of Control Engineering, Shahid Beheshti University, Tehran, Iran.