

Optimized Performance of Consensus algorithm in Multi Agent System Using PSO

Safanah Mudheher Raafat

Control and System Eng.Dept., University of
Technology, Baghdad - IRAQ

60154@uotechnology.edu.iq

Teaba Wala Aldeen Khairi

Computer Science Dept., University of Technology,
Baghdad - IRAQ

twk232004@yahoo.com

Received: 28-Sep.-2017 **Revised:** 03-Dec.-2017 **Accepted:** 12-March-2018

<http://doi.org/10.29194/NJES21020292>

Ahmed Mudheher Hasan

Control and System Eng.Dept., University of
Technology, Baghdad - IRAQ

60163@uotechnology.edu.iq

Karar Ghalib Ali

Control and System Eng.Dept., University of
Technology, Baghdad - IRAQ

karrar.g.ali@gmail.com

Abstract

This paper provides a theoretical framework for analysis of consensus algorithm for multi-agent networked systems considering the role of directed information flow. Improvement of the performance of the implemented consensus algorithm has been achieved by using Particle Swarm Optimization (PSO). Concepts of information consensus in networks and methods of convergence are applied as well. Our analysis framework is based on tools algebraic Graph Theory (GT). Simulation of multi-agent system and the performance of a consensus algorithm have been discussed. Acceleration the network while approaching the required goal has been accomplished and elimination of undesired swing that appears during the acceleration was proved.

Keywords: Consensus algorithm, Graph theory, Multi-agent network system, Particle swarm optimization.

1. Introduction

Distributed coordinated control of multi-agent robotics had recently attracted research interests; both in control theory and robotics. Various applications of cooperative control as for the spacecraft's, mobile robots, UAVs, sensor networks and other areas such as optimization can be found in literature [1].

Consensus algorithm is a key problem in formation control for multi-agent systems. The goal of that algorithm is to develop network topology and distributed protocol that allows the attainment of intricate goals. Consensus algorithm of multi-agent systems depends analytically on keeping a connected communication network among the agents [2].

The distributed consensus algorithm can be used to reach agreement between the agents where it computes the average of an initial set of measurements [2]. The calculation of the average value is done through local information exchange between neighbors. The speed of the consensus

algorithm to reach the average value depends on the states of each agent such as the position of the agent.

Early research on consensus algorithm focuses on fixed topologies, where the communication links and the nodes on a network are assumed constant throughout an area. A study on the consensus algorithm with random network was presented by [3] where the authors proposed pairwise gossip defined as that every two neighbored nodes can update their states at each iteration, and so on till all nodes reach the consensus value. However, this approach acts slowly since it requires storing all the data after each iteration. Additionally, large memory is needed. Meanwhile, L. Xiao and S. Boyd [4, 5] proposed a method to change the weight of each edge between vehicles (i.e. agents) that is required to apply the consensus algorithm for it. The aim is to find the optimal weights required to decrease the convergence time. The algorithm was treated as linear iteration procedure. Unfortunately, for a large network control system, the application of this approach may be unfeasible, since a new topology has to be acquired after each change in the communication between sensors under given computational constraints.

For large-scale graph, Olfati-Saber *et al.*, [6] proposed a "random rewiring" to increase the convergence speed in consensus algorithm. However, in some applications, changing the topology may be physically difficult and even costly. Ruozi Sun *et al.*, [7] suggested changing the weights of the graph through dependence on Euclidian area by using special algorithm called Weighted Dynamic Topology Control (WDTC) algorithm. Nevertheless, this algorithm is more complex and has no regard to the performance of network. In addition, Junghun Ryu *et al.*, [8] used Borel Cayley graphs to minimize the distance between vehicles in the network, which may lead to an increase in the number of edges between vehicles.

Optimization methods had been adopted to efficiently manage multi agent problems, like in

[9], where the implementation of cascade-connectivity swarm optimization algorithm to solve multi-objective optimization problem in order to stabilize networking layer. In addition, a discrete-time stochastic extremum seeking algorithm is used to convergence to a Nash equilibrium in [10].

In this paper, the concept of graph theory was implemented to construct the communication topology among different agents within a multi-agent system. Moreover, a robust optimization algorithm based on Particle Swarm Optimization (PSO) is utilized to improve the performance of the consensus algorithm in order to reach the optimum state for each agent in the system.

2. Background

2.1 Principals of Agent and Multi Agent System

The basic definition of an agent is “a system component with specific properties”. An agent is any entity that senses its environment and acting over it. An agent includes sensory inputs, goal, action and domain knowledge[11].A general single-agent framework is illustrated in Fig. 1.

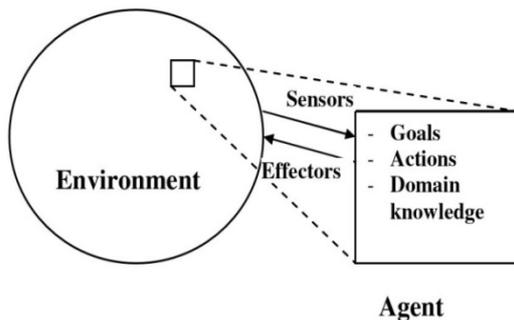


Figure (1): A general single-agent framework [11].

Reactivity, Deliberative, Sociality, and Autonomy are the most important properties of an agent [6]. Fig. 2 illustrates reactive and deliberative agents.

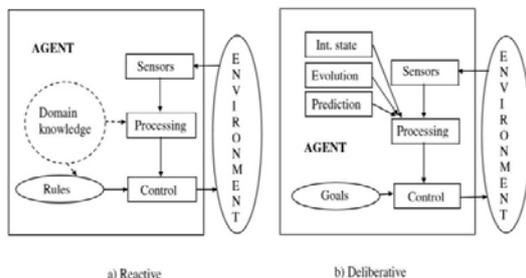


Figure (2): Reactive and deliberative agents [11].

Multi-agent system (MAS) can be defined as a group of agents who cooperate and communicate with each other to perform a specific task. MAS can treat complex problems that can't be treated

with a single agent. The simplest MAS is a multi-vehicle [12].

When a single agent in MAS has its own goal independently of the others, this is called discrete MAS. However, in this system there is no need for the cooperation and communication between them. When all agents have the same goal or solving the same problem, all agents are needed to cooperate to do this job through communication and cooperation between them [13].

2.2 Spectral Graph Theory

In this section, properties and some necessary definitions of spectral graph theory will be reviewed.

Assume that we have a network with N nodes, which called *vertices* and communication lines between them called edges, the graph topology can be represented as $G\{V,E\}$ in which the *vertices* $V=(1,\dots,N)$ and the edges $E \subseteq V \times V$.

There are two types that define the flow state between nodes; directed graph if the direction given otherwise undirected graph when there is no direction between nodes.

The model of communication topology in MAS is very important to exchange the information between agents. The graph theory can be used to model this topology meanwhile the digraph (i.e. directed graph) can be used to direct the information state of each agent to another agent. In order to apply consensus algorithm the digraph should be directed spanning tree model. Moreover, There are two important roles in directed spanning tree that should accomplished to model the communication between agents in MAS [14]:

- **First rule: the directed spanning tree should be strongly connected.**
- **Second rule: the directed spanning tree should be balanced.**

Definition 1:

The *adjacency* matrix A of G represents the communication between each node with its neighbors while the adjacency matrix of a G with n nodes is the $n \times n$ matrix where the i, j element is equal to 1 if node i and node j are neighbors, and zero otherwise [15].

$$a_{ij} = \begin{cases} 1 & \text{if } e_{ij} \in E \\ 0 & \text{if else} \end{cases} \dots (1)$$

a_{ij} Represent the entries of the *adjacency* matrix.

e_{ij} Represent edge between node i and node j

Definition 2:

For weighted graph the *laplacian* matrix calculated as in following equation [15]

$$l_{ii} = \sum_{j=1, j \neq i}^n a_{ij} \dots (2)$$

$$l_{ij} = -a_{ij} \dots (3)$$

Definition 3:

A vertex is balanced if its in-degree is equal to its out-degree. A directed graph is balanced if every vertex is balanced [15].

3. Consensus algorithms in a multi-agent system

Suppose there are i vehicles in a specific area and each vehicle has an information state. The information state defines the information that needed by each vehicle to be coordinated with another vehicle. This information may be vehicle position or velocity or oscillation phase and so on [16].

In consensus algorithm, the information states of each vehicle are updated and shared with another vehicle in the network. The updated information states are based on the information state of their neighbors. The consensus algorithm updates the information state of each vehicle and makes the information state of each vehicle in the network reach to the same value so that all vehicles have similar dynamics [17].

The most common continuous-time consensus algorithm is [17]:

$$\dot{x}_i(t) = - \sum_{j=1}^n a_{ij}(t) (x_i(t) - x_j(t)), \quad i = 1, 2, \dots, n \quad \dots (4)$$

where $a_{ij}(t)$ is the entry of the adjacency matrix of the graph at time t , and $x_i(t)$ is the information state of the i^{th} vehicle. When $a_{ij}(t)$ is equal to zero, that's mean there is no communication between vehicle i and vehicle j .

Continuous-time consensus algorithm in equation above can be written in matrix form as [18]

$$\dot{x}(t) = -L(t)x(t) \quad \dots(5)$$

where $L(t)$ is the digraph *laplacian* at time t and $x = [x_1, \dots, x_n]^T$.

Eq. (5) used with unbalanced graph theory. For balance graph, the following equation below will be used:

$$\dot{x} = -diag\{w\} Lx \quad \dots (6)$$

where $w = [w_1, w_2, w_3]^T$ is the positive left eigenvector of L that satisfy $w^T \mathbf{1} = 1$, where $\mathbf{1}$ is a vector of ones. The left columns of eigenvector w are entries of the diagonal matrix $diag\{w\}$ [19].

The continues time consensus algorithm can be achieved only when the *laplacian* matrix of digraph has zero simple eigenvalue [20].

4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is inspired by behavior of bird flocking. This algorithm consists of swarm of particles i.e. group of random particles where each single solution is a bird (particle) in the search space. Optimized solution for every particle is determined by fitness

function. Group of birds search for food by observing fitness function [21]. By following leader particle which is nearest to the food they can find the food. Leader particle is nothing but current optimal solution. So, every problem is initialized with random particles. PSO is based on birds swarm searching for optimal food sources in which direction of birds movement is influenced by its current movement, the best food source experienced by it ever and best food source any bird in the swarm ever experienced (i.e. known as personal best and global best values) and they get updated new best values after each iteration in PSO algorithm [22]. The personal best value is represented as u_p and global best value is represented as u_g . Particles movement is decided by following iteration in PSO as:

$$u_i(n+1) = u_i(n) + v_i(n+1) \quad \dots$$

$$n = 0, 1, 2, \dots, N-1 \quad \dots$$

(7)

where: u_i is the position of particle i , v_i is the velocity of particle i , n is number of iterations (initially $n = 0$), while N is total number of iterations.

The velocity of the particle is given as:

$$v_i(n+1) = v_i(n) + 2r_{1i}(n)[u_{pi}(n) - u_i(n)] + 2r_{2i}(n)[u_g(n) - u_i(n)] \dots (8)$$

where: u_p is the personal best position, u_g is the personal best position, $u_{pi}(n) - u_i(n)$ calculates vector in the direction of the personal best position and $u_g(n) - u_i(n)$ gives vector directed towards the global best position. r_{1i} and r_{2i} both represent random vectors which has values uniformly distributed between 0 and 1.

5 Consensus Performance in a Multi-Agent System

The performance of continuous time consensus with the multi-agent in a network is described in this section, where, the vehicle has been taken as an example for an agent. The continuous consensus algorithm has been applied two cases. The first case, the graph topology is considered not to be balanced; therefore, the application of the consensus algorithm will be applied using eq. (5). As a result, the states of the vehicle aren't reaching to the average state.

In the second case, graph theory is balanced; the application of the consensus algorithm will be according to eq. (6). In this case, the *Laplacian* matrix multiplied by left column of its eigenvector. That will change the unbalanced graph topology to balanced graph topology. The states of vehicles in this case will reach to the average state [23].

For illustration, three vehicles have been modeled in three different topologies, as shown in **Fig. 3**. The models of presented digraphs in (a)

and (b) are not strongly connected because not all agents have reached to the other agent, while in (c) it is strongly connected because each agent can reach to another agent.

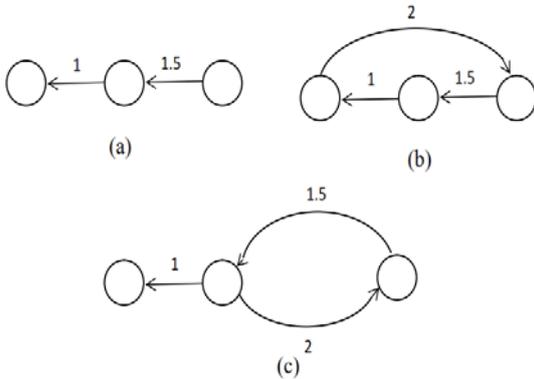


Figure (3): Three different communication topologies; (a) and (b) are not strongly connected, and (c) is strongly connected.

The adjacency matrices of each model in Fig. 3 are:

$$A(a) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1.5 \\ 0 & 0 & 0 \end{bmatrix} \quad \dots (9) A(b) =$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1.5 \\ 2 & 0 & 0 \end{bmatrix} \quad \dots (10)$$

$$A(c) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1.5 \\ 0 & 2 & 0 \end{bmatrix} \quad \dots (11)$$

From this adjacency, the *Laplacian* matrices can be found according to eq.(2) and eq. (3).

$$L(a) = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ 0 & 0 & 0 \end{bmatrix} \quad \dots (12)$$

$$L(b) = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ 0 & -2 & 2 \end{bmatrix} \quad \dots (13)$$

$$L(c) = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ -2 & 0 & 2 \end{bmatrix} \quad \dots (14)$$

Laplacian matrix plays a key role in the performance and convergence time of the consensus algorithm according to the *Laplacian* eigenvalues.

5.1 Simulation Example

Let's consider the communication models, as given in Fig. 3, where each of the *Laplacian* matrices $L(a), L(b)$ and $L(c)$ has simple zero eigenvalues, Hence, it can be shown that only the digraph with a directed spanning tree, has zero simple eigenvalue of *Laplacian* matrices. That means consensus can be achieved if and only if the digraph has a directed spanning tree.

If $\|x_i - x_j\| \rightarrow 0$ as $t \rightarrow \infty$ then the consensus can be achieved between vehicles [12].

As an example, the communication topologies described in Fig. 3 will be used with initial information states refer to the three vehicle

positions x_1, x_2 and x_3 as 0.2, 0.4 and 0.6 respectively. Each of the topologies in Fig. 3 has the common feature of containing a spanning tree graph theory. Although these topologies have zero simple eigenvalue, Fig. 3-a and Fig. 3-b are not strongly connected while the Fig. 3-c is strongly connected. Fig. 3-c will be used as communication topology in the following two cases.

In the first case, the common consensus algorithm of eq. (5) will be applied to the model in Fig. 3-c where the *Laplacian* matrix $L(c)$ is given in eq. (15). In this case, the consensus state is not achieved at the average consensus even though the digraph is strongly connected, because the graph is not balanced. Fig. 4 shows the simulation results of this case.

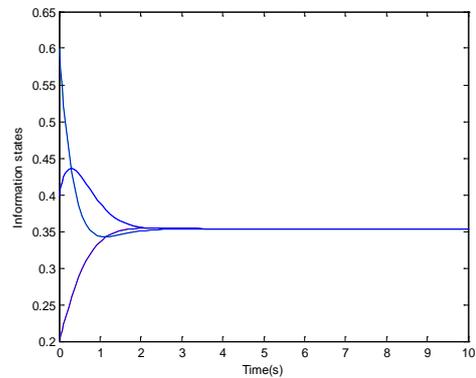


Figure (4): Simulation results corresponds to $\dot{x} = -L(c) x$

In the second case, eq. (6) has been used to change the unbalanced graph topology to balanced graph topology. In this case, the consensus is achieved at the average since the graph Γ [diag $\{w\} L3$] is balanced, as shown in Fig. 5.

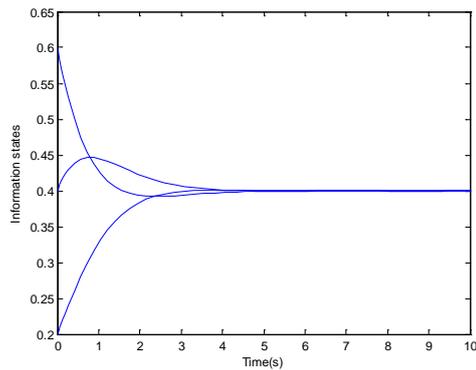


Figure (5): Simulation results corresponds to $\dot{x} = -\text{diag}\{w\} L(c) x$

The left column eigenvalue w of $L(c)$ is $\{-0.7682, -0.5121, -0.3841\}$, it can be verified $\dot{x} = -\text{diag}\{w\} L(c) x$ (15)

The solution to this equation according to that w is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = - \begin{bmatrix} -0.7682 \\ -0.5121 \\ -0.3841 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1.5 & -1.5 \\ -2 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \dots(16)$$

6 Improving the performance of the continuous consensus algorithm

From the simulation of case two in previous example (5-1), it can be seen that the required number of iterations to reach to the average consensus state is more than 5 iterations. In every iteration, each vehicle takes time from receiving information from its neighbor and processing that information to update the local state. Actually this long time is considerably impractical, especially in online or emergency applications. The acceleration of the consensus algorithm is more vital to many applications.

To improve the performance of a consensus algorithm in multi-agent system, we have added a new factor α to the consensus algorithm named as an acceleration factor. Therefore, eq. (16) can be rewritten as:

$$\dot{x} = -\alpha [\text{diag}\{w\} Lx], \quad \alpha \in \mathbb{R} \dots (17)$$

The acceleration factor can be tuned to reduce the required number of iterations when applying consensus algorithm for different number of vehicles in the network. Eventually, the performance of consensus algorithm would be improved. Nevertheless, unsuitable value of α causes “swinging” of the consensus state. The swing (which is a light changing of the consensus state) occurs after the information states of vehicles reach consensus. **Fig. 6** illustrates the resulting swing for the model, given in Example 5-1, that happens after reaching consensus for an acceleration factor α equal to 10.

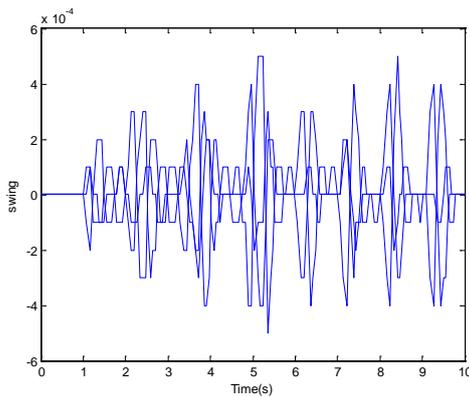


Figure (6): The swing of information states after reaching consensus, $\alpha=10$.

Thus, the acceleration factor has limited the undesired changes while reaching the consensus state.

7 Integration of the PSO method with consensus algorithm

In order to select minimum number of iterations as well as avoid swinging at target stage, PSO algorithm is utilized to select an optimal value for the proposed acceleration factor α . Moreover, the integrated method procedure is shown in **Fig. 7**.

The automatic tuning process will help to choose an optimal value of the acceleration factor α for different number of vehicles in the network, given the PSO parameters, as shown in **Table 1**. In addition, **Fig. 8** illustrates the performance of the applied consensus algorithm for the model given in subsection 5-1, with and without using PSO. The results illustrate the reduction in standard deviation when using PSO compared to that of stand-alone consensus algorithm, as shown in **Fig. 9**, where the Standard Deviation of error is computed for each case. The error between the target value for the state and its current value has been computed as follows:

$$\text{error} = |x_t(i, k) - x(i, k)|, i=1, \dots, n, k=1, \dots, N_i \dots(18)$$

where: x_t is the target value for a state, n is the number of states, N_i is the number of iterations.

Meanwhile, **Fig. 10** shows the elimination of swinging after using PSO.

7.1 Simulation Example:

The effect of the acceleration factor with the consensus algorithm can be seen with another model topology of four vehicles. **Fig. 11** shows the communication topology model of multi-vehicles and the information states are chosen as {0.2, 0.4, 0.6, 0.8}. The digraph in **Fig. 11** has spanning tree and the *Laplacian* matrix has zero simple eigenvalues.

The adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 2.5 \\ 2 & 0 & 0 & 0 \end{bmatrix} \dots (19)$$

The *Laplacian* matrix is

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1.5 & -1.5 & 0 \\ 0 & 0 & 2.5 & -2.5 \\ -2 & 0 & 0 & 2 \end{bmatrix} \dots (20)$$

Table 1: Parameters for PSO.

Parameters	Value
Fitness function	$Value = (\alpha - n)^2$
Swarmsize	10
Correction factor	2
Maximum iterations	10 ~ 50
Initial particle position	Input data
Best position so far	1000
Initial velocity	0
Inertia	1.0
Where: n is the number of agents - 1 α is the acceleration factor	

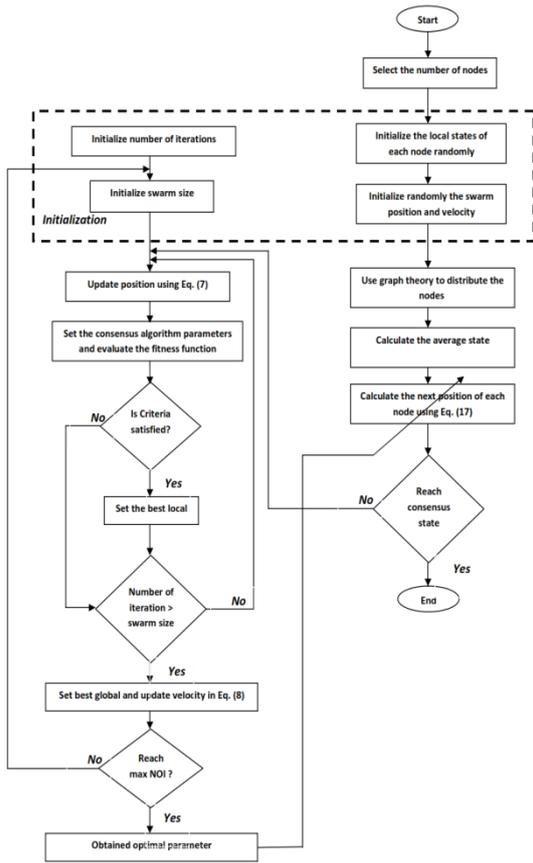


Figure (7): Flowchart for the integrated consensus algorithm in multi-agent system with PSO.

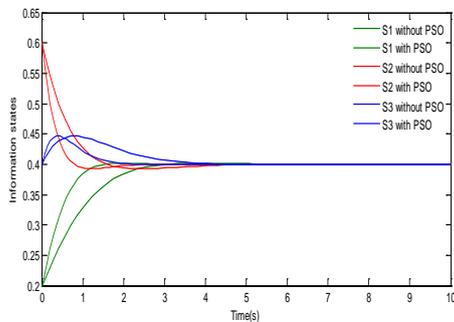


Figure (8): Simulation results using consensus algorithm with and without PSO.

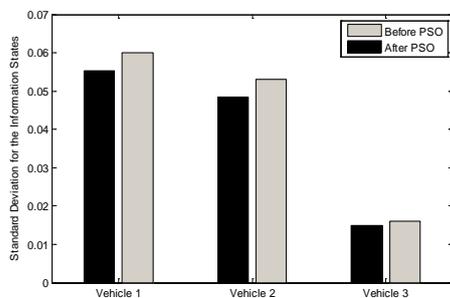


Figure (9): Comparison between consensus algorithm for three vehicles with and without PSO algorithm.

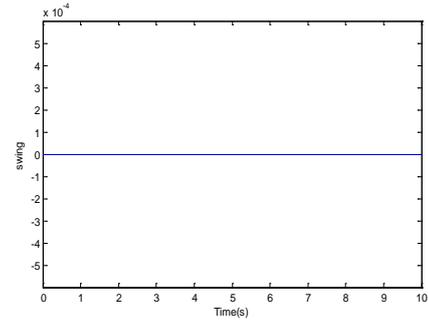


Figure (10): Swing with optimal α chosen by PSO, $\alpha = 2.0198$.

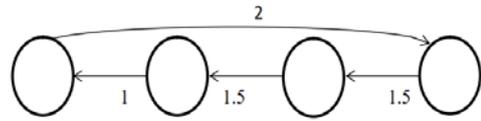


Figure (11): The communication topology between four vehicles.

Fig. 12 shows the simulation results of applying consensus algorithm both with and without optimized value of α that is obtained by PSO. Meanwhile, the reduction in standard deviation using PSO algorithm is shown in **Fig. 13**. The resulting swing when the best value of α is selected randomly is shown in **Fig. 14**, while **Fig. 15** shows the elimination of the swing when using optimal α by PSO.

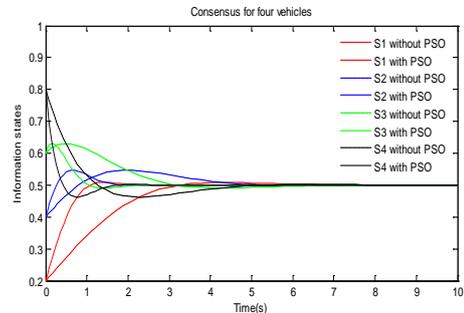


Figure (12): Performance of consensus algorithm applied for the model of four vehicles given in Fig. (11).

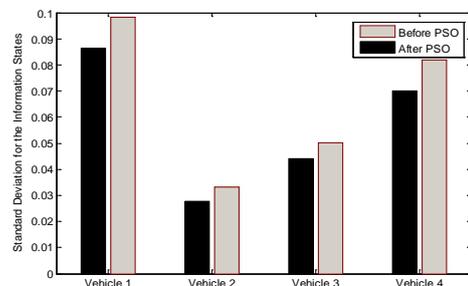


Figure (13): Comparison between consensus algorithm for four vehicles with and without PSO algorithm.

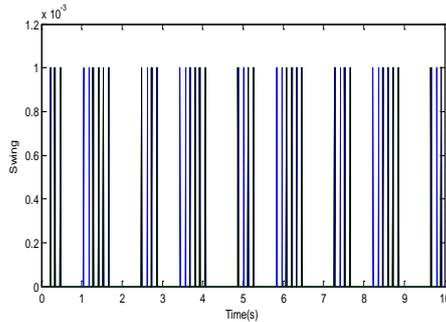


Figure (14): Swinging of four vehicles model as given in Fig. 11.

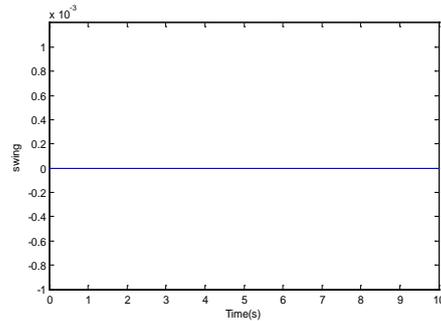


Figure (15): Swinging of four vehicles model as given in Fig. 11 with optimal value of α chosen by PSO, $\alpha = 3.1412$.

8 Discussion and Conclusions:

In this paper, different communication models between MAS have been built using direct graph theory. The first communication model consists of three vehicles (i.e. nodes) and the other model contains four vehicles. In order to accelerate consensus of MAS, the addition of an acceleration factor to the consensus algorithm has been proposed. Optimal value acceleration factor has been reached by PSO algorithm while eliminating swinging that may occur during acceleration. Finally, considerable reduction in Standard Deviation of the information states when using PSO was reached to 10% compared to that without using PSO for the three vehicles and 19% for the four vehicles.

References

[1] X. Wang, H. Su, X. fanWang, and G. Chen, "An overview of coordinated control for multi-agent systems subject to input saturation," *Perspectives in Science*, Vol. 7, pp. 133-139, 2016.

[2] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Trans. Automat. Contr.*, Vol. 54, No. 11, pp. 2506–2517, 2009.

[3] J. Coon, C. P. Dettmann, and O. Georgiou, "Impact of boundaries on fully connected random geometric networks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, Vol.

85, No. 1, pp. 1–5, 2012.

[4] S. Boyd, P. Diaconis, J. Sun, and L. Xiao, "Fastest mixing markov chain on a path," *Am. Math. Mon.*, Vol. 113, No. 1, pp. 70–74, 2006.

[5] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, Vol. 53, No. 1, pp. 65–78, 2004.

[6] R. O. Saber, J. a Fax, and R. M. Murray, "Consensus and Cooperation in Multi-Agent Networked Systems," *Proc. IEEE*, Vol. 95, No. 1, pp. 215–233, 2007.

[7] R. Sun, J. Yuan, I. You, X. Shan, and Y. Ren, "Energy-aware weighted graph based dynamic topology control algorithm," *Simul. Model. Pract. Theory*, Vol. 19, No. 8, pp. 1773–1781, 2011.

[8] J. Ryu, J. Yu, E. Noel, and K. Wendy Tang, "Borel Cayley Graph-Based Topology Control for Consensus Protocol in Wireless Sensor Networks," *ISRN Sensor Networks*, Vol. 2013, Article ID 805635, 15 pages, 2013. doi:10.1155/2013/805635

[9] Xianlin Zeng, Zhenyi Liu, and Qing Hui, "Energy Equipartition Stabilization and Cascading Resilience Optimization for Geospatially Distributed Cyber-Physical Network Systems", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 45, No. 1, pp. 25–43, January 2015.

[10] Milo's S. Stankovic', Karl H. Johansson, and Du'san M. Stipanovic, "Distributed Seeking of Nash Equilibria With Applications to Mobile Sensor Networks", *IEEE Transactions on Automatic Control*, Vol. 57, No. 4, pp.904-919, April 2012.

[11] M. Glavic, "Agents and Multi-Agent Systems: A Short Introduction for Power Engineers," , Technical Report, pp. 1–21, 2006.

[12] Y. S. and K. Leyton-Brown, Y. Shoham, and K. Leyton-brown, "Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations," *ReVision*, Vol. 54, No. 1-4, p. 513 p., 2008.

[13] P. Stone, P. Ave, and F. Park, "Multiagent Systems : A Survey from a Machine Learning Perspective," *Robotics*, Vol. 8, No. 3, pp. 345–383, 2000.

[14] M. B. Egerstedt and M. Egerstedt, "Graph-Theoretic Methods for Multi-Agent Coordination," *Robomat 2007*, pp. 1–10, 2007.

[15] S. Bessy, "Some problems in graph theory and graphs algorithmic theory," *Diss. Université Montpellier II-Sciences et Techniques du Languedoc*, 2012.

[16] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control: Collective group behavior through local interaction," *IEEE Control Syst. Mag.*, Vol. 27, No. 2, pp. 71–82, 2007.

- [17] Y. Zheng, J. Ma, and L. Wang, "Consensus of Hybrid Multi-agent Systems," IEEE Transactions on Neural Networks and Learning Systems, Vol. PP, No. 99, pp. 1-17, 2017.
- [18] L. Fang and P. J. J. J. Antsaklis, "Information Consensus of Asynchronous Discrete-time Multi-agent Systems," Proceedings of the 2005, American Control Conference, 2005, Vol. 3, pp. 1883-1888, 2005.
- [19] E. Atkins, W. Ren, R. W. Beard, and S. Member, "Information Consensus and its Application in Multi-vehicle Cooperative Control Information Consensus and its Applications in Multi-vehicle Cooperative Control," IEEE Control Systems Magazine, pp.71-82, April 2007.
- [20] J.-G. Dong, L. Qui, "Complex Laplacians and applications in multi-agent systems", 2015, [online], Available:<http://arxiv.org/abs/1406.1862v2>.
- [21] S. Jamali, R. Leila, and S. Gudakahriz, "An Energy-efficient Routing Protocol for MANETs: a Particle Swarm Optimization Approach," Journal of Applied Research and Technology, Vol. 11, No. 6, pp. 803-812, 2013.
- [22] T. Schoene, S. a Ludwig, and R. J. Spiteri, "Step-Optimized Particle Swarm Optimization," IEEE Congress on Evolutionary Computation , Brisbane, QLD, Australia, pp. 1-9, 2012
- [23] H. Du, S. Li, and S. Ding, "Bounded consensus algorithms for multi-agent systems in directed networks," Asian J. Control, Vol. 15, No. 1, pp. 282-291, 2013.

الأداء الأمثل لخوارزمية الإجماع في نظام متعدد الوكلاء باستخدام خوارزمية سرب الجسيمات الامثليه

كرار غالب علي
قسم هندسه السيطرة والنظم
الجامعه التكنولوجيه

طيبه ولاء الدين خيرى
قسم علوم الحاسوب
الجامعه التكنولوجيه

أحمد مظهر حسن
قسم هندسه السيطرة والنظم
الجامعه التكنولوجيه

سفانه مظهر رأفت
قسم هندسه السيطرة والنظم
الجامعه التكنولوجيه

الخلاصة

هذا البحث يقدم الاطار النظري لتحليل خوارزمية التوافق في الأنظمة الشبكية متعددة الوكلاء مع مراعاة دور تدفق المعلومات الموجه. وقد تم تحسين أداء خوارزمية التوافق التي تم تنفيذها باستخدام تحسين سرب الجسيمات (PSO). وتم تطبيق مفاهيم توافق الآراء في المعلومات في الشبكات وأساليب التقارب. ويستند إطار التحليل لدينا على أدوات نظرية الرسم البياني جبري (GT). وقد نوقشت محاكاة نظام متعدد الوكلاء ومناقشة أداء خوارزمية التوافق. تسريع الشبكة في حين يقترب من الهدف المطلوب قد أنجزت والقضاء على البديل غير المرغوب فيه الذي يظهر خلال تسارع.