2018 26th Mediterranean Conference on Control and Automation (MED)
Zadar, Croatia, June 19-22, 2018

Consensus Protocol for Underwater Multi-Robot System Using Two Communication Channels

Barbara Arbanas[1], Tamara Petrovic[1], Stjepan Bogdan[1]

Abstract— **In this paper, we apply a simple consensus protocol to a simulated underwater multi-robot system using two communication channels. The approach is motivated by a system in which robots, called aMussels, are dispersed in the environment and measure its properties over longer periods of time. Each unit is equipped with two underwater communication devices: *green light*, short-range communication device based on modulated light, and *nanomodem*, long-range acoustic communication device. Except for range, the two communication channels differ in the transmission speed. Both are combined in order to attain fast and accurate average consensus. We bring the results of the protocol applied to the simulated system and analyze its convergence.**

I. INTRODUCTION

In the recent decades, in an ever-growing field of robotics, more attention has been given to multi-robot systems (MRS). As technology advances, robots are becoming smaller, cheaper and more diverse in their applications. Therefore, there is a rise in research in the field of self-organizing networked multi-agent systems, so-called *swarms*. Since they usually consist of many units, the main challenge is to devise algorithms that can control this emerging type of systems.

Consensus algorithms in the control of multi-agent systems are particularly compelling for their simplicity and a wide range of applications. Foundation of consensus protocols in multi-agent systems lies in the field of distributed computing. In networks of agents, *consensus* means to "reach an agreement regarding a certain quantity of interest that depends on the state of all agents" [1]. A consensus algorithm (protocol) is a series of rules that define information exchange between an agent and all of his neighbors on the network.

Recently, there has been a surge of interest among scientists from different fields in problems related to consensus in networked multi-agent systems. Some of the most interesting problems include the collective behavior of flocks and swarms [2]–[4], formation control for MRS [5]–[7], optimization-based cooperative control [8], sensor fusion [9], synchronization of coupled oscillators, asynchronous distributed algorithms, and many more.

We are particularly interested in the application of consensus algorithms to underwater systems. Specifically, to an underwater multi-robot system for intelligent long-term monitoring of underwater ecosystems devised within the subCULTron* project. This robotic system is comprised of three types of agents: aPads - surface vehicles, aFish - underwater vehicles and aMussels - vehicles attached to the sea-ground. This paper focuses on control algorithms for aMussels, specially devised sensor hubs used to monitor environmental properties of the sea. They are the largest subgroup in the subCULTron swarm, with the total count of 100-120 units. There are many foreseen decisions aMussels will have to make during their operation, such as identifying sensor failures, dividing into subgroups and managing their energy balance (when to charge). Considering a large number of units and challenges of the underwater environment, we consider applying consensus algorithms for their resolution.

There have not been many advancements in the area of underwater consensus protocols. To the best of our knowledge, the only underwater consensus applications include formation control of tethered and untethered underwater vehicles [10]–[12] and tracking of underwater targets using acoustic sensor networks (ASN) [13]. Both of those approaches have been validated only in the simulation environment, using ideal communication channels.

One of the main challenges in such a system is operation under the conditions of poor communication. In the subCULTron system, each aMussel is equipped with two underwater communication devices: *green light*, short-range communication device based on modulated light, and *nanomodem*, long-range acoustic communication device. Another important distinction between the two is much faster transmission speed of the *green light* device. In this work, we consider the interplay of the two in a specially devised protocol for reaching average value consensus. The protocol was formulated in our previous work [14], where we considered a group of aMussels communicating using only acoustic scheduled communication. We validated the method both in simulation, as well as in an experimental testbed. The need for inventing a new protocol came from the nature of the medium in which the communication takes place. Since each device has to have a dedicated time interval to emit messages in order to avoid interference, all communication is scheduled and the protocol updates variable values in a discrete manner. We further build upon that work by introducing another communication channel, with a goal of speeding up the convergence of the procedure, which is the main contribution of this paper.

The paper is organized as follows. In the following section we outline the consensus protocol for scheduled commu-

This paper was supported by the EU-H2020-FETPROACT-2-2014 project subCULTron - Submarine Cultures Perform LongTerm Robotic Exploration of Unconventional Environmental Niches, grant agreement No. 640967.

[1]Laboratory for Robotics and Intelligent Control Systems (LARICS), Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia – barbara.arbanas@fer.hr

*http://www.subcultron.eu/

nication, and detail its application to two communication channels. Section III entails the simulation results of the method applied to the underwater robotic system. Finally, we give the conclusion in the Section IV.

II. CONSENSUS PROTOCOL FOR SEQUENCED COMMUNICATION

We consider a distributed system of n integrator agents with dynamics $\dot{x}_i = u_i$, where $x_i \in \mathbb{R}$ represents agent i's state, whereas u_i is the control input. The goal of consensus protocols, in general, is to design a control algorithm that enables agents to reach a homogeneous stationary state $x^* \in \mathbb{R}$, that is,

$$x_1 = x_2 = \ldots = x_n = x^*. \tag{1}$$

State variables x_i can correspond to some physical property. Initial states of all agents are represented by a vector $\mathbf{x_0} = [x_1(0)\ldots x_n(0)]^T$.

Agents update their values based on the information exchanged via local communication with their neighbors. The underlying communication graph is defined as $G = (V,E)$ with the set of nodes $V = 1,2,\ldots,n$ and edges $E \subseteq V \times V$.

An iterative form of the discrete-time consensus algorithm for reaching the average value of n integrator agents [1] is stated as follows:

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in N_i} (x_j(k) - x_i(k)), \tag{2}$$

where k and $k+1$ denote discrete time steps of the algorithm, and $N_i = \{j \in V : (i,j) \in E\}$ represents a set of all nodes adjacent to the ith node, and $\varepsilon \in \langle 0,1]$ is the step-size.

The collective dynamics of the network under this algorithm can be written as

$$x(k+1) = Px(k), \tag{3}$$

with $P = I - \varepsilon L$. I is the identity matrix and L is known as *graph Laplacian of* G. Matrix L is defined as

$$L = D - A, \tag{4}$$

where $D = diag(d_1,\ldots,d_n)$ is the degree matrix of G with elements $d_i = \sum_{j \neq i} a_{ij}$ and zero off-diagonal elements. $A = [a_{ij}]$ is the adjacency matrix of the graph G.

Consensus algorithm (2) assumes synchronous information exchange at every time step k with no time delays, and its convergence into the average value of all initial states is mathematically proven [1]. Although some algorithms take into account systems with delayed and asynchronous communication [15]–[17], there are none to the authors' knowledge that operate under the scheduled sequenced communication.

A. Consensus protocol for sequenced communication

Let's consider a communication scheme in which agents take turns transmitting messages in a sequential manner. A simple illustration of such a protocol is given by Figure 1. T represents the period of message transmission, while the actual duration of each transmission is given by T_{di}, where $i \in \mathbb{N}$ represents the index of the allocated time slot.

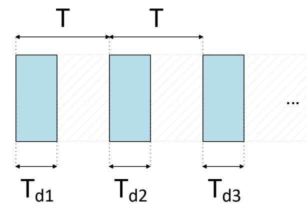


Fig. 1. Communication scheme for scheduled sequenced communication

Generally, $T_{di} \neq T_{dj}$ but in order to avoid interference, we must ensure that $T_{di} < T \; \forall i$. Assuming the transmission of the first message starts at a time t_1, beginning of ith transmission is given by $t_i = t_1 + (i-1)T$.

Each time slot can, therefore, be defined by its start and duration as (t_i, T). Assignment of those slots to particular agents is arbitrary but can greatly affect the performance of the algorithm. This is not the focus of this paper, so we opt for the simplest solution, which is sequential assignment repeated cyclically (*round-robin* [18]).

We model the described behavior through manipulating Laplacian matrix to switch the communication topology depending on the current state of the system. Modified equation (3) is given by

$$x(k+1) = P'x(k), \tag{5}$$

with $P' = I - \varepsilon L'$, where L' represents the graph Laplacian calculated as $L' = D' - A'$. Both D' and A' are calculated using a specially devised vector m used as a *mask* of the underlying communication topology. Assuming communication scheme defined earlier and round-robin scheduling of time slots in a system of n agents, elements of vector $m = (m_j)$ are defined as

$$m_j(t) = \begin{cases} 1, & t = t_1 + (i-1)T \\ 0, & otherwise \end{cases}, \tag{6}$$

where $j = i \mod n$, and i represents the index of the time slot. Each change of the vector m triggers the execution of the algorithm described with (5). Degree and adjacency matrices are defined as

$$D' = diag(Am),$$
$$A' = A \; diag(m).$$

B. Implications of the protocol to two communication channels

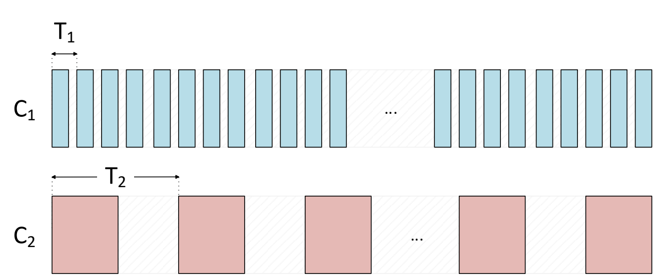


Fig. 2. Communication scheme for two different communication channels

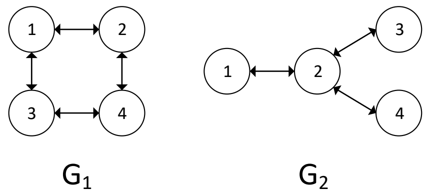


Fig. 3. Example system of 4 agents with two communication channels

As stated before, we apply the described protocol to the system that uses two communication channels, which differ both in range, as well as transmission speed. Since a hard constraint on both channels is to avoid emitting from two devices at the same time, we construct a simple communication scheme, as depicted in Figure 2. Basically, we have two parallel independent communication protocols with transmission intervals T_1 and T_2.

The collective dynamics of the network under this protocol are given by

$$x(k+1) = \left(I - \varepsilon\left(L_1' + L_2'\right)\right)x(k). \tag{7}$$

L_1' and L_2' are defined as in the case of a single communication channel. This algorithm can be generalized to n communication channels as

$$x(k+1) = \left(I - \varepsilon \sum_{i\in(1,n)} L_i'\right)x(k). \tag{8}$$

Example of the protocol run on a simplified system: Consider a system of 4 agents with two different communication channels defined by graphs G_1 and G_2 in Figure 3. Therefore, corresponding adjacency matrices A_1 and A_2 are

$$A_1 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad A_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Communication schemes for channels C_1 and C_2 are defined by $T_1 = 1s$, $T_2 = 5s$, and $t_{11} = t_{12} = 0$. Here we run several steps of the algorithm with step size $\varepsilon = 0.5$.

1) $k = 0,\ t = 0,\ x(0) = \begin{pmatrix} 0 & 10 & 20 & 30 \end{pmatrix}^T$
   $m_1 = \mathbf{0},\ m_2 = \mathbf{0}$
2) $k = 1,\ t = 1s$
   $m_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T,\ m_2 = \mathbf{0}$

$$\begin{aligned} x(1) &= \left(I - \varepsilon\left(L_1' + L_2'\right)\right)x(0) \\ &= \left(I - \varepsilon\left(D_1' - A_1' + D_2' - A_2'\right)\right)x(0) \\ &= (I - \varepsilon\,(diag(A_1 m_1) - A_1 diag(m_1) + \\ &\qquad diag(A_2 m_2) - A_2 diag(m_2)))\,x(0) \\ &= \left[I - 0.5\left[diag\left(\begin{pmatrix} 0 & 1 & 1 & 0 \end{pmatrix}^T\right) - \right.\right. \\ &\qquad \left.\left.\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\right]\right] x(0) \\ &= \left(I - \begin{pmatrix} 0 & 0 & 0 & 0 \\ -0.5 & 0.5 & 0 & 0 \\ -0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\right)x(0) \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} x(0) \\ &= \begin{pmatrix} 0 & 5 & 10 & 30 \end{pmatrix}^T \end{aligned}$$

3) $k = 2,\ t = 2s$
   $m_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}^T,\ m_2 = \mathbf{0}$

$$\begin{aligned} x(2) &= \begin{pmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0.5 & 0 & 0.5 \end{pmatrix} x(1) \\ &= \begin{pmatrix} 2.5 & 5 & 10 & 17.5 \end{pmatrix}^T \end{aligned}$$

4) $k = 3,\ t = 3s$
   $m_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}^T,\ m_2 = \mathbf{0}$

$$\begin{aligned} x(3) &= \begin{pmatrix} 0.5 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0.5 & 0.5 \end{pmatrix} x(2) \\ &= \begin{pmatrix} 6.25 & 5 & 10 & 13.75 \end{pmatrix}^T \end{aligned}$$

5) $k = 4,\ t = 4s$
   $m_1 = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^T,\ m_2 = \mathbf{0}$

$$x(4) = \begin{pmatrix} 6.25 & 9.375 & 11.875 & 13.75 \end{pmatrix}^T$$

6) $k = 5,\ t = 5s$
   $m_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T,\ m_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T$

$$\begin{aligned} x(5) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} x(4) \\ &= \begin{pmatrix} 6.25 & 6.25 & 9.0625 & 13.75 \end{pmatrix}^T \end{aligned}$$

7) $k = 6,\ t = 6s$
   $m_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}^T,\ m_2 = \mathbf{0}$

$$x(6) = \begin{pmatrix} 6.25 & 6.25 & 9.0625 & 10 \end{pmatrix}^T$$

8) $k = 7,\ t = 7s$
   $m_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}^T,\ m_2 = \mathbf{0}$

$$x(7) = \begin{pmatrix} 6.25 & 6.25 & 9.0625 & 10 \end{pmatrix}^T$$

$$\ldots$$

In the following section, we apply the outlined method to a simulated system, modeling previously mentioned short- and long-range communication channels as C_1 and C_2 in Figure 2, respectively.

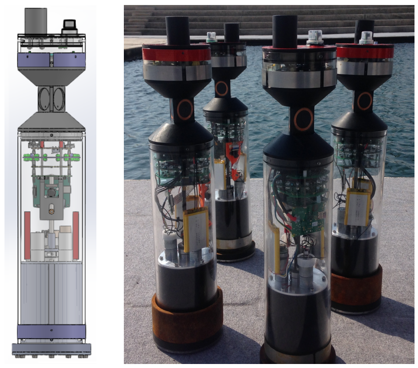


Fig. 4. aMussel robot – on the left is the model of the robot, while right side of the figure shows a photo of aMussels next to the pool during the trials.

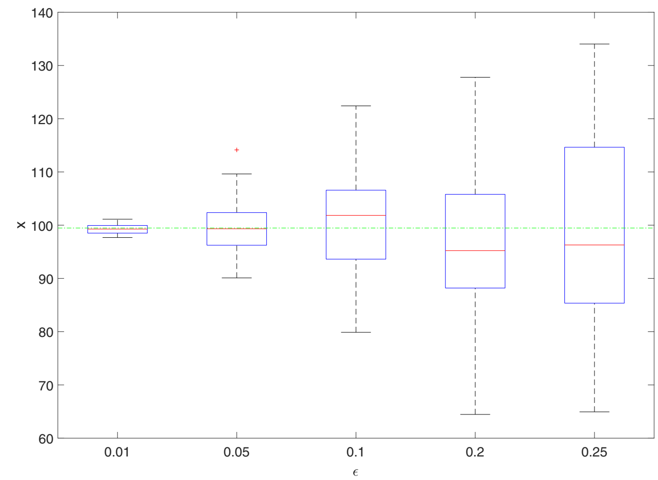


Fig. 6. Influence of the step size ε to the calculated consensus value x^*. For each setup 50 simulations with randomized initial states $\mathbf{x_0}$ were performed.

III. SIMULATION RESULTS

We consider a system comprised of 15 aMussels (Figure 4), all equipped with both short- and long-range communication devices. The reach of long-range communication is such that all agents can communicate with each other at all times. Short-range communication, however, splits agents into several local groups. The topology of short-range communication is depicted in Figure 5.

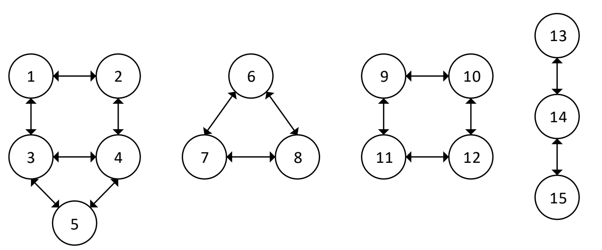


Fig. 5. Topology of the BL communication channel.

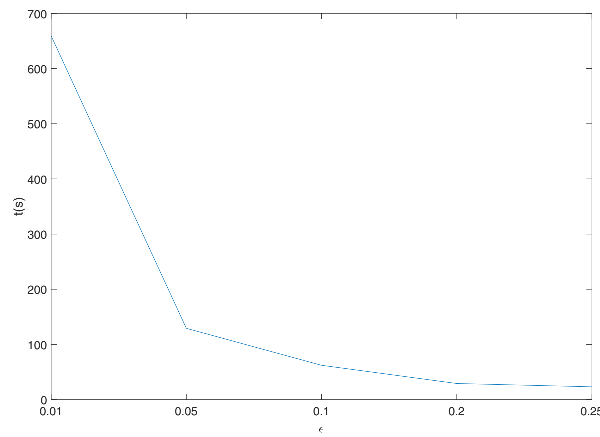


Fig. 7. Influence of the step size ε to the convergence time ($T_2 = 1s$).

Here we first, in short, analyze the influence of step size ε to the procedure using only long-range communication and then include the short-range communication to observe its effects on the proposed method.

A. Influence of the step size to convergence

One of the most important parameters influencing method's accuracy and speed is the step size ε. Figure 6 shows how it relates to the accuracy of the solution. Clearly, smaller step sizes ensure greater precision, while choosing too large parameter renders solutions inaccurate. This is due to a much greater influence of the information received at the beginning of the procedure since algorithm executes sequentially with each information update. Therefore, the order in which agents communicate plays a big role in consensus value. This effect can be alleviated by choosing smaller algorithm step size.

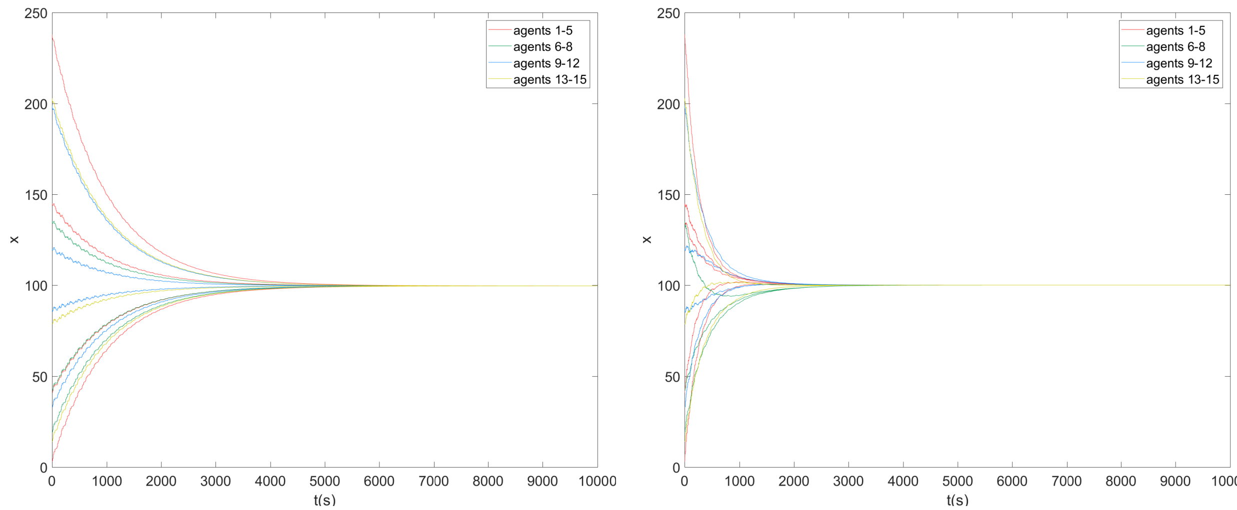
On the other side, it is interesting to observe how the same parameter choice influences convergence times (Figure 7). We consider the system to have converged when the standard deviation of all agents' states in the previous 10 steps is less than 0.1. A significant drop in convergence time with the increase of ε is noticeable. In this setup we used communication period $T_2 = 1s$.

In a sense, a compromise between procedure speed and accuracy can be made by choosing the appropriate ε. For our application scenario, where dynamics of the processes are very slow, we choose the smallest and the most precise ε and we use it in the continuation of the analysis.

B. Two communication channels

Next, we consider a setup in which both communication channels are active in information exchange. Their respective transmission intervals are set to $T_1 = 1s$ and $T_2 = 10s$. These parameters are chosen to be the most similar to what is possible in the envisioned application system.

State trajectories of agents of the system for one algorithm run with step size $\varepsilon = 0.01$ are shown in Figure 8. Different agent groups are marked with distinct colors on the plot, as stated in the legend. For comparison, left-hand side graph shows trajectories for execution of the algorithm using only the long-range communication and it exhibits the behavior expected of a typical consensus algorithm. On the other side,



(a) Not using short-range communication

(b) Using short-range communication

Fig. 8. State trajectories of 15 agents. Algorithm parameters are set to $T_1 = 1s$, $T_2 = 10s$ and $\varepsilon = 0.01$.

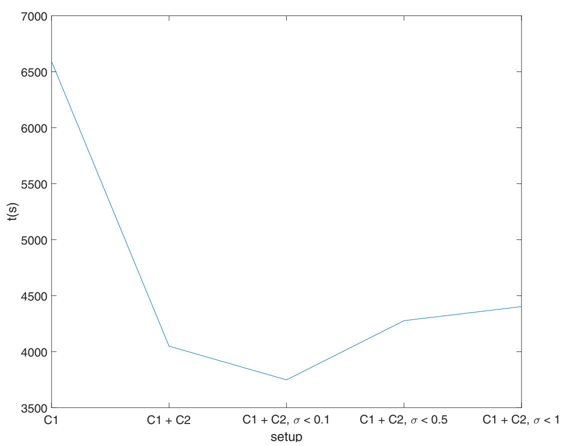


Fig. 9. Convergence time with regard to different strategies used (without short-range communication ($C1$), using short-range communication without turning it off ($C1+C2$), and with turning it off at different convergence stages within subgroups ($C1+C2$ with $\sigma < 0.1$, $\sigma < 0.5$, and $\sigma < 1$).

in Figure 8(b) we can notice much faster convergence and by observing more closely, convergence among subgroups can be distinguished. For example, early group convergence can be noticed on an example of the second subgroup (green on Figure 8(b)). An agent with the highest initial state has first taken a steep downward trajectory to approach its' group's average value since information flow within groups is much faster. Upon arrival of more messages from non-group members, the whole group changes the trajectory to meet the global consensus value.

The final thing we consider is the effectiveness of the proposed method compared to using only one communication channel. Results on Figure 9 show a significant increase of algorithm speed by utilizing both communication channels, as opposed to using only long-range communication.

Also, since convergence among distinct groups occurs much faster than global convergence, we consider turning off short-distance communication channel as soon as satisfactory convergence level has been reached. This is done with a goal of saving energy used to transmit messages. The measure of group convergence is standard deviation σ of group members' values and it is stated on the x axis of the graph on Figure 9. An interesting phenomenon is an increase of the convergence speed if the short-range communication is turned off near convergence within groups ($\sigma < 0.1$). At this stage of the algorithm group influence benefits to the convergence are diminished and even prolong convergence into a global consensus value.

IV. CONCLUSION AND FUTURE WORK

In this paper, we devised a simple consensus protocol to a simulated underwater multi-robot system using two communication channels. The approach is motivated by an underwater multi-robot system for intelligent long-term monitoring of underwater ecosystems devised within the subCULTron project. A protocol is applied to a simulated system of stationary robots equipped with two underwater communication devices: *green light*, short-range communication device based on modulated light, and *nanomodem*, long-range acoustic communication device. Except for range, the two communication channels differ in the transmission speed. Both are combined in order to attain fast and accurate average consensus.

As a future work, we plan to deepen mathematical notation of the procedure and work on proving the convergence of the method. Also, we will experimentally validate the procedure on a real robotic system, both in a laboratory testbed, as well as in realistic conditions in the Venice lagoon. Future development of the proposed method to allow more diverse applications than just calculating an average value of the network state will be considered.

REFERENCES

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[2] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.

[3] R. Olfati-Saber and P. Jalalkamali, "Coupled distributed estimation and control for mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2609–2614, Oct 2012.

[4] R. A. Freeman, P. Yang, and K. M. Lynch, "Distributed estimation and control of swarm formation statistics," in *2006 American Control Conference*, June 2006, pp. 7 pp.–.

[5] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, Sept 2004.

[6] Z. Qu, J. Wang, and R. A. Hull, "Cooperative control of dynamical systems with application to autonomous vehicles," *IEEE Transactions on Automatic Control*, vol. 53, no. 4, pp. 894–911, May 2008.

[7] W. Ren, "Consensus strategies for cooperative control of vehicle formations," *IET Control Theory Applications*, vol. 1, no. 2, pp. 505–512, March 2007.

[8] M. Alighanbari and J. P. How, "Decentralized task assignment for unmanned aerial vehicles," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec 2005, pp. 5668–5673.

[9] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *2007 46th IEEE Conference on Decision and Control*, Dec 2007, pp. 5492–5498.

[10] M. A. Joordens and M. Jamshidi, "Consensus control for a system of underwater swarm robots," *IEEE Systems Journal*, vol. 4, no. 1, pp. 65–73, March 2010.

[11] V. Putranti, Z. H. Ismail, and T. Namerikawa, "Robust-formation control of multi-autonomous underwater vehicles based on consensus algorithm," in *2016 IEEE Conference on Control Applications (CCA)*, Sept 2016, pp. 1250–1255.

[12] M. Mirzaei, N. Meskin, and F. Abdollahi, "Robust distributed consensus of autonomous underwater vehicles in 3d space," in *2016 International Conference on Control, Decision and Information Technologies (CoDIT)*, April 2016, pp. 399–404.

[13] J. Yan, B. Pu, X. Tian, X. Luo, X. Li, and X. Guan, "Distributed target tracking using consensus-based bayesian filtering by underwater acoustic sensor networks," in *2017 36th Chinese Control Conference (CCC)*, July 2017, pp. 8789–8794.

[14] B. Arbanas, T. Petrovic, and S. Bogdan, "Consensus protocol for underwater multi-robot system using scheduled acoustic communication," in *To appear in the proceedings OCEANS 2018 IEEE/MTS*, 2018.

[15] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, Sept 2008.

[16] J. Qin, C. Yu, and S. Hirche, "Stationary consensus of asynchronous discrete-time second-order multi-agent systems under switching topology," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 986–994, Nov 2012.

[17] L. Fang and P. J. Antsaklis, "Information consensus of asynchronous discrete-time multi-agent systems," in *Proceedings of the 2005, American Control Conference, 2005.*, June 2005, pp. 1883–1888 vol. 3.

[18] W. Stallings, *Operating Systems: Internals and Design Principles*, 6th ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2008.