Average Consensus-Based Data Fusion in Networked Sensor Systems for Target Tracking

Md Ali Azam
Electrical Engineering
South Dakota School of Mines and Technology
mdali.azam@mines.sdsmt.edu

Shawon Dey
Electrical Engineering
South Dakota School of Mines and Technology
shawon.dey@mines.sdsmt.edu

Hans D. Mittelmann
School of Mathematics and Statistical Sciences
Arizona State University
mittelmann@asu.edu

Shankarachary Ragi
Electrical Engineering
South Dakota School of Mines and Technology
Shankarachary.Ragi@sdsmt.edu

***Abstract*—Decentralized and distributed autonomous sensing over networked sensor systems has many applications in surveillance, Internet of Things (IoT), autonomous cars, and UAV swarms tactics. In this study, we develop an average consensus-based decentralized data fusion approach for a target tracking application. Specifically, we extend the standard average consensus algorithm to merge the local state estimate information with that of the neighbors. We test the performance of our consensus based data fusion approach for various network configurations. We also perform numerical studies to compare the performance of our approach against the standard Bayesian data fusion approach.**

***Index Terms*—Networked sensor systems, Decentralized average consensus, Sensor data fusion, Target tracking**

I. Introduction

Autonomous and adaptive sensing has applications such as target tracking, surveillance [1], autonomous car navigation [2], and UAV swarm tactics [3], [4]. Particularly, target tracking via adaptive sensing is becoming increasingly important in autonomous car industry for accurate pedestrian detection and tracking [5]. Sensors such as RADAR, LIDAR, optical sensors, thermal sensors are typically used to measure the target state including its position, velocity, and acceleration. Target tracking with multiple sensors was studied in the past, e.g., [3], where a central fusion node was responsible for making sensing decisions (e.g., sensor location - assuming sensor mounted on a UAV) for all the sensors combined. Clearly, sensing decisions optimized for all the sensors combined provides the best target tracking performance as these decisions are coupled via sensor data fusion. The main drawback with these centralized decision making methods is that they are computationally intensive as the computational complexity is exponential in the decision space and the number of sensors. To address this challenge, we investigated decentralized strategies in the past to some extent [4].

This work was supported in part by Air Force Office of Scientific Research under grant FA9550-19-1-0070.

In this study, we develop a decentralized autonomous sensing method over a networked sensor system for a target tracking application. Specifically, we extend an existing approach called *average consensus algorithm* to perform decentralized data fusion while tracking a moving target. The sensor network is modeled by an undirected graph, which is assumed to be non-time varying. Each sensor generates a noisy measurement of the target state. The presence of an edge between the nodes or sensors means that the sensors are allowed to exchange information/messages for data fusion. In this study, we assume that each sensor maintains a local tracker (or tracking algorithm, e.g., Kalman filter), which updates its local target state estimate using the locally generated sensor measurements and the information it receives from its neighbors. We measure the performance of the above consensus algorithm with *average target tracking error* - the mean-squared error between the target state (ground truth) and the estimate. As a benchmark, we also implement the standard Bayesian data fusion approach for performance comparison.

The authors of [6] have surveyed both classical approaches and recent advances in multi-sensor data fusion and consensus filter for sensor networks. The authors of [7] reviewed the key theories and methodologies of distributed multi-sensor data fusion and discussed their advantages like graceful degradation, scalability, and interchangeability. *Average consensus* was studied previously in distributed computing [8] and for achieving consensus among agent values (a real number possibly representing its opinion or state). In [9], a distributed consensus algorithm was developed for obtaining the averages of the node data over networks with large volume of data. N. Gupta et. al. proposed an asynchronous distributed average consensus algorithm [10] to guarantee information-theoretic privacy in multi-agent systems. In [11], the authors provide a theoretical framework for analysis of consensus algorithms for multi-agent networked systems. In [12], the authors developed a distributed consensus tracking filter to solve the target tracking problem. The authors in [13] discussed algorithms for solving decentralized consensus optimization problems.

978-1-7281-3783-4/20/$31.00 ©2020 IEEE

A. Key Contributions

- We extend the *average consensus* algorithm [9] to track a moving target via a decentralized network of sensors. We compare the performance of this method against a standard benchmark method - *decentralized Bayesian data fusion approach* [7].
- We perform a numerical study to quantify the impact of various sensor network configurations (e.g., varying degrees of the nodes) on the performance of the *average consensus* algorithm.

The rest of the paper is organized as follows. Section II presents the problem specification and the objectives. Section III provides the problem formulation and the methods followed by the simulation results in Section IV. Finally, we provide concluding remarks and future scope in Section V.

II. Problem Specification

In our study, we assume there are n sensors tracking a moving target in a decentralized setting, where the sensors are connected via an undirected graph. The target is assumed to be moving on a 2-D plane, where the motion is modeled via a stochastic process, i.e., the state-transition law is a linear model with zero-mean Gaussian noise. We assume the sensor measurement law is also linear with zero-mean Gaussian noise. Thus, each sensor maintains and updates a local target state estimate via Kalman filtering algorithm.

We assume that the sensors have limited battery power and computational capabilities, which sets limitations on the sensors in terms of how they generate measurements and communicate with other sensors. Specifically, we assume that the sensors can either sense (generate target measurements) or exchange information with neighboring sensors, but not simultaneously.

Communications: The sensors have communications capabilities, i.e, each sensor can transmit or receive data to/from the sensors they share edges in the network graph. We further assume that the communications delay is negligible.

Sensor network: The n sensors are assumed to be connected via an undirected graph. Each sensor i has a set of neighbors, denoted by $N(i)$, where sensor $j \in N(i)$ if there is an edge connecting j with i.

Performance measure: We measure the performance of the algorithms using *average tracking error*, which is the mean-squared error between the target state and the estimates averaged over all the sensors and over time.

Objective: The objective is to compare the performance the *average consensus* algorithm against the standard *decentralized Bayesian data fusion* technique for target tracking with a decentralized sensor network. We measure the performance of these algorithms for different sensor network configurations.

III. Problem Formulation

A. Tracking Approach

In our study, $\{1, ..., n\}$ represent the sensor indices, and S_i represents the 2D location of sensor i. The target's motion is described by a linear state-space model (specifically *constant velocity* model [14]):

$$x_k = Ax_{k-1} + \theta_k, \quad \theta_k \sim \mathcal{N}(0, Q) \tag{1}$$

where x_k is the state of the target at time k (which includes the target's 2D location, 2D velocity, and 2D acceleration), A is a state transition matrix, and θ_k is process noise with zero-mean normal distribution with co-variance matrix Q. Sensor i generates a position measurement z_k^i given by:

$$z_k^i = Hx_k + v_k^i \tag{2}$$

where H is the observation matrix given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

which means that the sensors only generate positional measurements. Here $v_k^i \sim \mathcal{N}(0, R(x_k, S_i))$ is the random measurement noise modeled as a zero-mean normal distribution, where the co-variance matrix $R(x_k, S_i)$ captures the dependence of the noise characteristics on the location of the target with respect to the sensor. Here, R_k reflects 10% range uncertainty and 0.01π radian angular uncertainty. Since the state and the observation laws are linear with zero-mean Gaussian noise disturbances, we run Kalman filter at each sensor node to maintain and update the target state posterior distribution with mean and co-variance given by $\hat{x}_{k|k}^i$ and $P_{k|k}^i$.

Clearly, if the sensors do not exchange any information, the tracking performance suffers at each node. The sensors are connected via an undirected graph, where the presence of an edge between nodes i and j means that the sensors are allowed to exchange information. So, we extend an approach called *average consensus* algorithm to allows sensors to exchange information in a manner that improves the target tracking performance across the sensor network.

B. Average Consensus

Average consensus algorithms let a network of sensors or agents reach a common consensus on certain attributes (real numbers) such as the agent opinions, sensor measurements, etc. Specifically, in these approaches, each agent or sensor updates/replaces (in an iterative manner over time) its local value by taking a weighted average between its local value and the values from all the neighbors. We extend this approach to let the sensors in our problem reach a common consensus on their state estimate parameters (mean vector and covariance matrix). Let y_k^i is a vector obtained by concatenating $\hat{x}_{k|k}^i$ and $P_{k|k}^i$ into a column vector at sensor i at time k. $N(i)$ is the set of neighbors for i^{th} sensor. Average consensus algorithm applied to our problem is captured by the following equation:

$$y_{k+1}^i = \frac{\alpha y_k^i + (1-\alpha)\sum_{j \in N(i)} y_k^j}{\alpha + |N(i)|(1-\alpha)}, \quad \forall i \tag{3}$$

where α is a weighting parameter.

This algorithm achieves its objective if all the sensors reach consensus on the state estimation parameters, i.e., $y_k^i = y_k^j$ for all i, j.

C. Decentralized Bayesian data fusion

Multi-sensor data fusion techniques can be applied in both centralized and decentralized settings. In our study, we use decentralized Bayesian data fusion techniques over the sensor network. Each sensor has a local state estimate x_k^i which is updated in each time step by fusing x_k^i with the estimates from its neighboring sensors as given by the following equations (using standard Bayes rules [15]).

$$P_{k+1}^i = \left((P_k^i)^{-1} + \sum_{j=1}^{N(i)} (P_k^j)^{-1} \right)^{-1} \tag{4}$$

$$\hat{x}_{k+1}^i = P_{k+1}^i \left((P_k^i)^{-1} \hat{x}_k^i + \sum_{j=1}^{N(i)} (P_k^i)^{-1} \hat{x}_k^j \right)^{-1} \tag{5}$$

IV. Simulation Results

We implement our methods for a scenario with 10 sensors, i.e., $n = 10$. We set $\alpha = 0.5$ in the following numerical studies except when we evaluate the performance of our algorithms with varying α. We compare the performance of the average consensus algorithm against the decentralized Bayesian data fusion approach for different sensor network configurations with *average tracking error* (defined earlier) as the performance measure. In our numerical studies, we use error bars with one standard deviation to show the spread of the performance measure for multiple network graphs generated from a given configuration as discussed below (examples of configurations in Fig. 1).

Configuration I. This corresponds to a network where each sensor has the same degree, where the degree is given by D, which is referred to as *network degree*. We generate a random graph with n sensors and D network degree.

Configuration II. In this configuration, we generate a random graph with *edge probability* P_e, where P_e represents a probability of an edge existing between two sensors. We start with n sensors with no edges at the beginning, and we create an edge between every pair of sensors with probability P_e. We repeat this process until we get a connected network.

Configuration III. This corresponds to a network with a total number of edges N_e in a connected network.

As sensors typically have limited computational capability and limited battery life, we assume they can run only tracking algorithm while generating sensor measurements or only communicate with neighbors, i.e., run the consensus or data fusion methods as described in Section II. Specifically, in our study, sensors track the target for M time steps and apply the consensus/data fusion algorithms in the next M time steps, and repeat the process. During the M time steps when the consensus/data fusion algorithms are being applied, sensors update the state estimates of the target without the measurements, i.e., perform only prediction step and ignore the measurement update step. In other words, the uncertainty in the target state estimate steadily increases during these M time-steps.

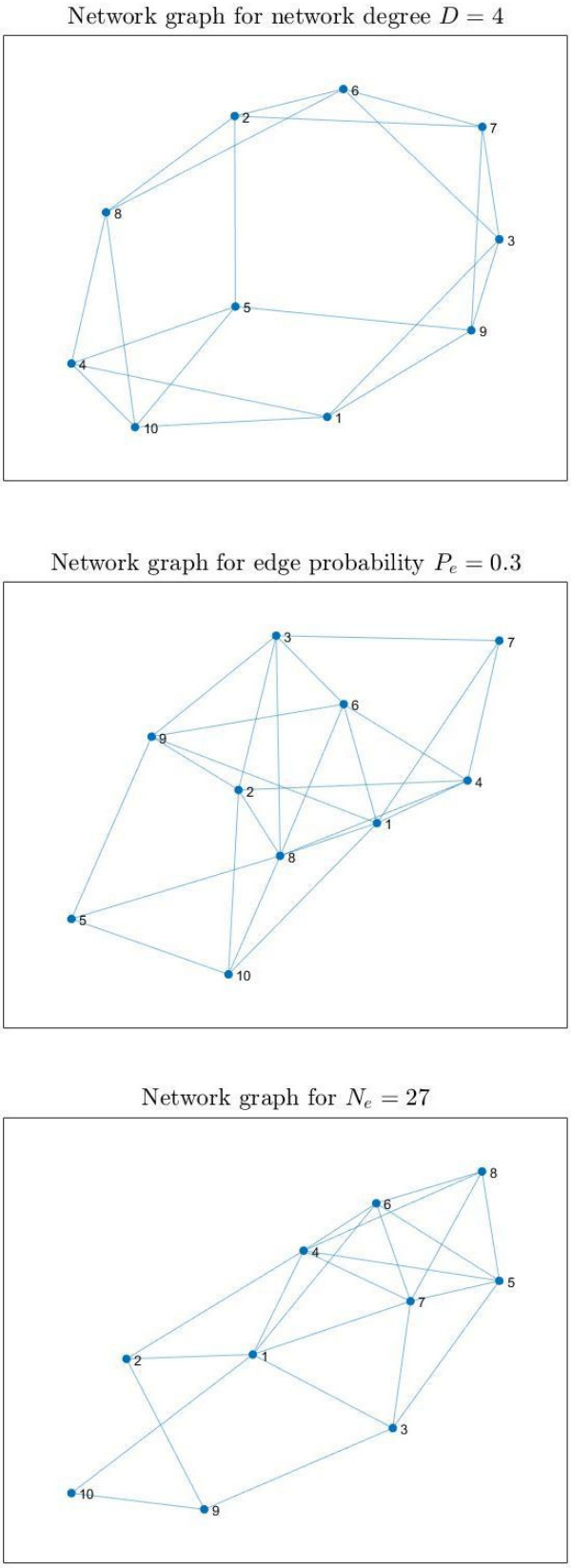

Fig. 1. Examples of configurations (configuration I, II, III from top to bottom)

Let Z represent the total number of time steps in our simulation run time. We set $Z = 300$ in this study. We define the *average tracking error* measure as follows:

$$\frac{1}{Z}\frac{1}{n}\sum_{k=1}^{Z}\sum_{i=1}^{n}\left\|\hat{x}_k^i - x_k\right\|_2^2$$

where x_k represents the ground truth at time k, and $\|\cdot\|_2$ is the Euclidean norm.

A. Average tracking error vs. M

We now compare the performance of *average consensus* and *decentralized Bayesian data fusion* algorithms for different values of M on five randomly generated graphs for $n = 10$. We evaluate the average tracking error, as defined earlier, for each value of M considered. Fig. 2 shows the average tracking error as a function of M, where $M \in \{3, 6, 9, \ldots, 24\}$. The figure suggests that the average consensus algorithm outperforms the data fusion approach for all values of M considered. The consensus algorithm seems to be more effective in merging information from multiple sensors than the standard decentralized Bayesian data fusion approach.

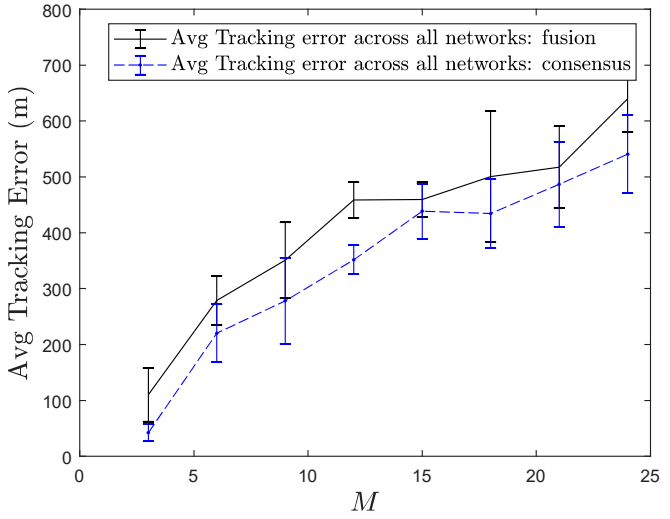

Fig. 2. Average tracking error across all sensors with respect to number of time steps, M

Fig. 3 represents average tracking error as a function of M for $M \in \{1, 2, \ldots, 9\}$. Fig. 3 shows that the average consensus and decentralized Bayesian data fusion algorithm give better performance for $M = 2$ and $M = 3$ respectively compared to all other values of M considered here.

B. Average tracking error for configuration I

We now evaluate the average tracking error as a function of the network degree as shown in Fig. 4. We compare the performance of these two algorithms on five randomly generated graphs for $M = 1$ and $n = 10$. We observe that the performance of both algorithms increase as the network degree increases. Furthermore, from Fig. 4, we observe that the average consensus algorithm performs better than the decentralized Bayesian data fusion method. This is an expected behavior since with greater network degree, the sensors have better capability in merging information from other sensors.

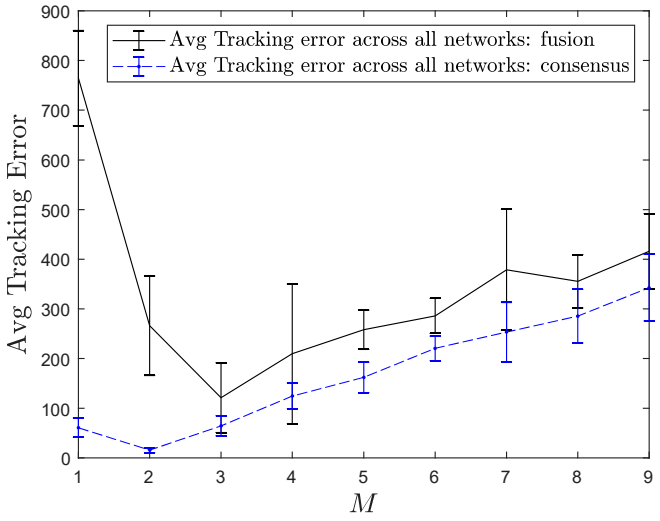

Fig. 3. Average tracking error across all sensors with respect to number of time steps, M

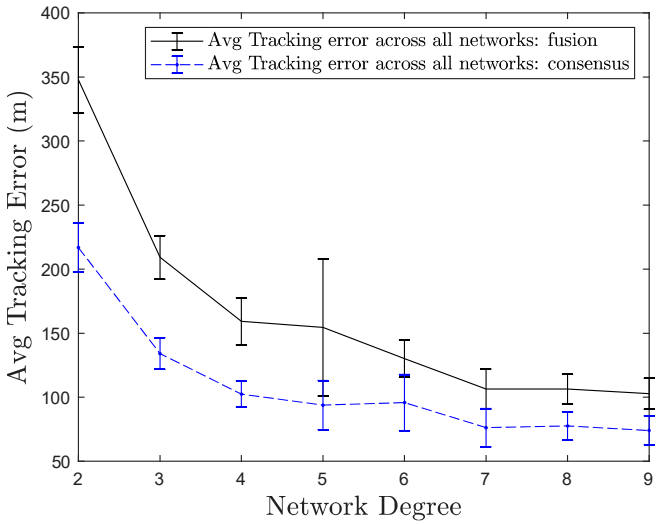

Fig. 4. Average tracking error across all sensors for configuration I

C. Average tracking error for configuration II

We now perform the same numerical study for a randomly generated graph by using Configuration II with different values of P_e drawn from the set $\{0.1, 0.2, \ldots, 1\}$. For each P_e, we generate 10 graphs. Fig. 5 shows that, for both algorithms, the average tracking error decreases with respect to P_e, which is expected since the network connectivity increases with increasing P_e. We also notice that the consensus algorithm outperforms the decentralized Bayesian data fusion approach for each P_e.

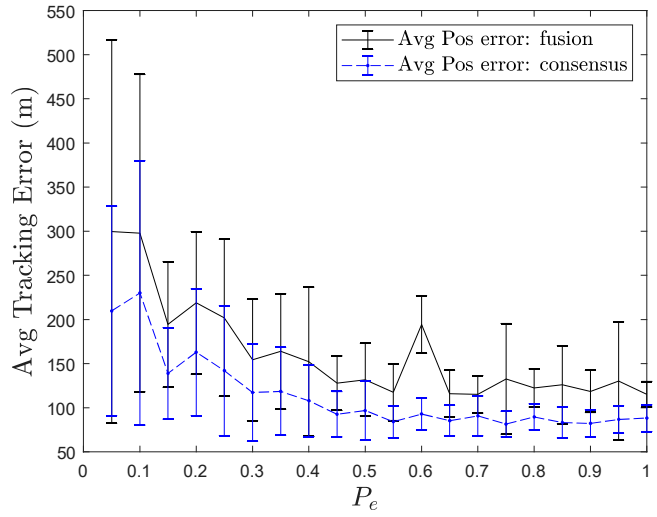

Fig. 5. Average tracking error across all sensors with respect to edge probability P_e

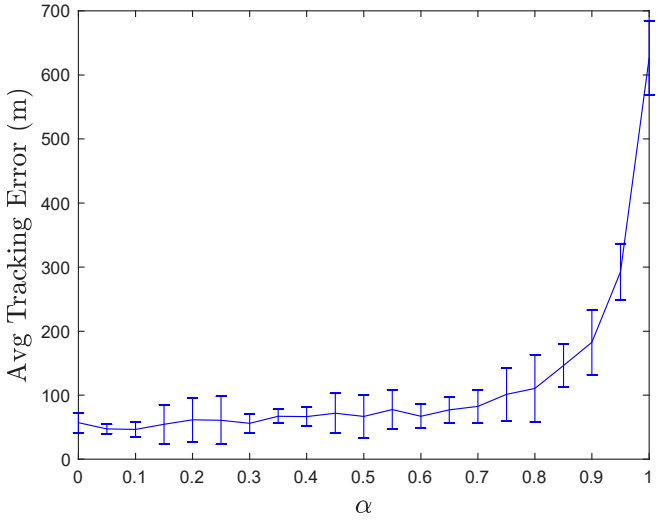

Fig. 7. Average tracking error across all sensors with respect to weighting parameter α

D. Average tracking error for configuration III

We now evaluate the average tracking error for different value of N_e as shown in Fig. 6. We generate (randomly) five graphs with Configuration III for this study. We observe that with increasing N_e, the performance of both of the algorithms increases. We fit 5^{th} degree polynomial curves for the performance plots in Fig. 6, which characterize the variation of the performance of the algorithms as a function of N_e.

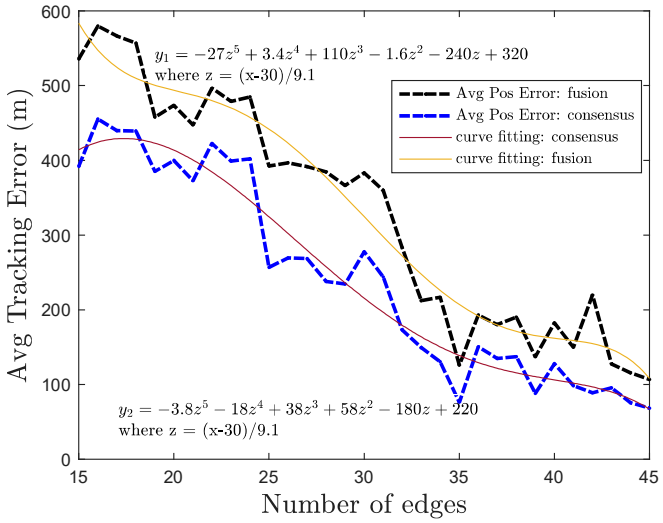

Fig. 6. Average tracking error across all sensors with respect to number of edges

E. Average tracking error for weighting parameter α

In this part, we study the performance of the average consensus algorithm with respect to the weighting parameter α. Here, $\alpha = 0$ means that the consensus algorithm replaces the local sensor's state estimate with the average of its neighbors' estimates. On the other hand, $\alpha = 1$ means that the consensus algorithm ignores the estimates from the neighbors and simply retains the local state estimate. For different values of α in the interval $[0, 1]$, we evaluate the average tracking error, as shown in Fig. 7. The figure shows that the average tracking error increases significantly when the value of α is close to 1.

V. Conclusion and Future Scope

In this study, we extended the *average consensus* algorithm for decentralized data fusion over a networked sensor system for target tracking. We studied the performance of our extended average consensus algorithm numerically for different network configurations and compared with standard Bayesian decentralized Bayesian data fusion as a benchmark. We found that the average consensus algorithm outperformed the decentralized Bayesian data fusion for all network configurations considered in this study. In the future, we will consider decentralized data fusion over time-varying sensor networks and develop graph-theoretic solutions to maximize the data fusion performance.

References

[1] C. Carthel, S. Coraluppi and P. Grignan, "Multisensor tracking and fusion for maritime surveillance," *2007 10th International Conference on Information Fusion*, Quebec, Que., 2007, pp. 1-6.

[2] A. Manzanilla, S. Reyes, M. Garcia, D. Mercado and R. Lozano, "Autonomous Navigation for Unmanned Underwater Vehicles: Real-Time Experiments Using Computer Vision," in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1351-1356, April 2019.

[3] S. Ragi, E. K. P. Chong, "Dynamic UAV path planning for multitargte tracking," *American Control Conference,* pp. 3845-3850, 2012.

[4] S. Ragi, E. K. P. Chong, "Decentralized Guidance Control of UAVs with Explicit Optimization of Communication," *J Intell Robot Syst*, vol. 73 pp. 811-822, 2014.

[5] S. Blackman and R. Popoli, "Design and analysis of modern tracking systems," Boston, USA: Artech House, 1999.

[6] W. Li, Z. Wang, G. Wei, L. Ma, J. Hu and D. Ding, "A Survey on Multisensor Fusion and Consensus Filtering for Sensor Networks," *Discrete Dynamics in Nature and Society*,Volume 2015, Article ID 683701, 2015.
[7] M. A. Bakr, S. Lee, "Distributed Multisensor Data Fusion under Unknown Correlation and Data Inconsistency," *Sensors*, Oct. 2017.
[8] N. A. Lynch, Distributed Algorithms. San Francisco, CA: Morgan Kaufmann, 1997.
[9] T. C. Aysal, M. J. Coates and M. G. Rabbat, "Distributed Average Consensus With Dithered Quantization," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4905-4918, Oct. 2008.
[10] N. Gupta, J. Katz, and N. Chopra. "Statistical Privacy in Distributed Average Consensus on Bounded Real Inputs." *arXiv preprint* arXiv:1903.09315 (2019).
[11] R. Olfati-Saber, J. A. Fax, R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215-233, Jun. 2007.
[12] S. Zhu, C. Chen, W. Li, B. Yang, X. Guan, "Distributed Optimal Consensus Filter for Target Tracking in Heterogeneous Sensor Networks," *IEEE Transactions on Cybernetics*,Volume: 43 , Issue: 6 , Dec. 2013.
[13] A. Nedich, A. Olshevsky, and W. Shi, "Decentralized Consensus Optimization and Resource Allocation," Lecture Notes in Mathematics, Springer Verlag, Vol. 2227, pp. 247287, 2018 (contributions of 2017 LCCC Workshop).
[14] X. R. Li and Y. Bar-Shalom, "Design of an interacting multiple model algorithm for air traffic control tracking," in *IEEE Transactions on Control Systems Technology*, vol. 1, no. 3, pp. 186-194, Sept. 1993.
[15] J. K. Hackett and M. Shah, "Multi-sensor fusion: a perspective," Proceedings., *IEEE International Conference on Robotics and Automation*, Cincinnati, OH, USA, 1990, pp. 1324-1330 vol.2