AFRL-RI-RS-TR-2019-005

DISTRIBUTED DETECTION AND CONTROL OF UNEXPECTED/EMERGENT BEHAVIORS IN MULTIAGENT SYSTEMS

BRADLEY UNIVERSITY

JANUARY 2019

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2019-005 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
GENNADY STASKAVICH
Work Unit Manager

/ S /
JULIE BRICHACEK
Chief, Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
JANUARY 2019	FINAL TECHNICAL REPORT	APR 2015 – JUL 2018

4. TITLE AND SUBTITLE

DISTRIBUTED DETECTION AND CONTROL OF UNEXPECTED/EMERGENT BEHAVIORS IN MULTIAGENT SYSTEMS

5a. CONTRACT NUMBER

5b. GRANT NUMBER FA8750-15-1-0143

5c. PROGRAM ELEMENT NUMBER 62788F

6. AUTHOR(S)

Jing Wang

5d. PROJECT NUMBER S2MA

5e. TASK NUMBER VV

5f. WORK UNIT NUMBER BR

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Bradley University, Department of Electrical and Computer Engineering
1501 West Bradley Avenue
Peoria, IL 61625

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RISC
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER

AFRL-RI-RS-TR-2019-005

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The study of dynamic behaviors of multiagent systems poses significant challenges due to the fact that complex system behaviors emerge as a result of their individual self-operating (sensing and actuating) capability, as well as of the interactions among agents. Some of those challenges include the comprehensive understanding of interaction mechanism among agents and the development of distributed resilient algorithms for detecting, designing, and controlling interaction dynamics and interaction topologies of multiagents. In this project, we studied fundamental issues from distributed estimation to distributed control of multiagent emergent behaviors. This multiple-year project leads to training of a number of graduate and undergraduate students and results in twenty-five research publications in the internationally refereed journals and conferences. Specifically, the highlights of the research outcomes include the following developments: 1) Designed a distributed estimation algorithm for the detection of certain global characteristic signals in multiagent systems, which can serve as feature indicators for the group collective behaviors. The event triggering mechanism for information transmission was further employed to reduce the communication load. 2) Designed a distributed sensor fusion algorithm for environmental monitoring by wireless sensor networks (WSNs) with limited communication.

15. SUBJECT TERMS
Distributed estimation, distributed control, distributed optimization, multiagent systems

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**GENNADY STASKAVICH**
U	U	U	UU	56	**19b. TELEPHONE NUMBER (*Include area code*)** **N/A**

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1.0 SUMMARY

Multiagent systems have found a wide range of potential applications in surveillance and reconnaissance, cooperative exploration for search and rescue missions, environmental sensing and monitoring, and cooperative transportation. It has been long recognized that multiagents may exhibit emergent dynamic behaviors due to local and intermittent interactions among individual agents. Unfortunately, the study of dynamic behaviors of multiagent systems poses significant challenges due to the fact that complex system behaviors emerge as a result of their individual self-operating (sensing and actuating) capability, as well as of the interactions among agents. Some of those challenges include the comprehensive understanding of interaction mechanism among agents and the development of distributed resilient algorithms for detecting, designing, and controlling interaction dynamics and interaction topologies of multiagents.

To address these challenges, we assembled a multi-investigator team from two universities and carried out research on fundamental issues from distributed estimation to distributed control of multiagent emergent behaviors. This multiple-year project leads to training of a number of graduate and undergraduate students and results in twenty-three research publications in the internationally refereed journals and conferences. Specifically, the highlights of the research outcomes include the following developments: 1) Designed a distributed estimation algorithm for the detection of certain global characteristic signals in multiagent systems, which can serve as feature indicators for the group collective behaviors. The event-triggering mechanism for information transmission was further employed to reduce the communication load. 2) Designed a distributed sensor fusion algorithm for environmental monitoring by wireless sensor networks (WSNs) with limited communication. The proposed algorithm relies on local information exchange among neighboring sensor nodes and estimation for a key left eigenvector of the communication matrix. The algorithm convergence can be ensured under the assumption that the communication topology among sensors is directed and strongly connected. 3) Developed a resilient distributed detection algorithm for multiple targets. 4) Designed adaptive cooperative control for a class of uncertain nonlinear multiagent systems. 5) Developed an approximate distributed gradient estimation method for networked system optimization. 6) Conducted extensive simulation and experimental validation on the proposed distributed algorithms by using Q-Bot2 mobile robots, Kilobots, AgentFly simulator, and Crazyflie Quadrotors.

2.0 INTRODUCTION

With the rapid development of computing, communication and sensing technology, recent years have seen an ever increasing research interest in the study of distributed multiagent systems. In particular, multiagent systems have found a wide range of potential applications in surveillance and reconnaissance, cooperative exploration for search and rescue missions, environmental sensing and monitoring, and cooperative transportation. One of the salient characteristics of multiagent systems is that local behaviors of individual agents may lead to certain emergent global behaviors through intermittent interactions among agents. These ubiquitous phenomena are often observed in biological swarms, such as flocks of birds, schools of fish, herds of animals, and colonies of bacteria [19, 23, 20]; in social networks, such as rhythmic applause, opinion dynamics, and decision making in animal groups [6]; in engineered systems, such as robotic networks [10, 2, 21], power grids [15], computer networks [17], and sensor networks [14], to name but a few.

In this project, the overall objective is to enhance the understanding of emergent behaviors in distributed multiagent systems through rigorous analysis and design for modeling, detection, estimation, and control of interaction dynamics and interaction topologies.

By focusing on this overall objective, we developed an integrated model-based approach for distributed detection and control of unexpected/emergent behaviors in multiagent systems. Specifically, we studied multiagents that are modeled as a network of dynamical systems, and through inter-agent sensing/communication, the multiagent systems exhibit emergent behaviors (by nature or by design) and accomplish coordinated tasks as a group. Such a problem setting fits into many realistic safety-critical multiagent systems, such as coordination of multiple UAVs. While significant progresses have been made in the design and analysis of multiagent systems under limited sensing/communication topologies [18, 7, 22, 11, 10, 1, 12], there are very limited results available in the literature to address issues on how collective behaviors emerge as well as on how to ensure the desired multiagent system behaviors through design and control. With the aid of rigorous tools from systems and controls theory as well as learning and adaptation methods, we studied the resulting emergent behaviors of multiagent systems by considering effects of both *interaction dynamics* and *interaction topologies*, and developed an efficient, predictable, and safe detection and control scheme to deal with unexpected/emergent behaviors in distributed multiagent systems.

This multiple-year project has rendered significant research outcomes from three main aspects: (1) development of distributed estimation algorithms for multiagent interaction topologies and multiagent dynamical behaviors; (2) development of distributed optimization and control algorithms for multiagent systems under limited information exchange and modeling uncertainties; (3) Implementation and validation of the proposed distributed algorithms through computer simulations and experimental testing on UAV and UGV platforms. The project also leads to training of a number of graduate and undergraduate students and results in twenty-three research publications in the internationally refereed journals and conferences.

The rest of this report is organized as follows. Section 3.0 presents the basic methods, assumptions and procedures in this research, and formulates the distributed estimation and control problem. Sections 4.0, 5.0, and 6.0 present the technical results and discussions on three sets of results in term of distributed estimation algorithms, distributed optimization and control algorithms, and experiments, respectively. Section 7.0 concludes the report.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

The research in this project was carried out based on the assumption that the prototype multiagent systems are defined by a set of dynamical equations given below

$$\dot{z}_i = \mathcal{F}_i(z_i, v_i) + \Delta\mathcal{F}_i(z_i), \quad y_i = H_i(z_i), \tag{1}$$

where $i \in \{1, \cdots, N\}$ is the index for agent i and there are N agents in the group, $z_i \in \Re^{n_i}$ is the state, $v_i \in \Re^m$ is the control (interaction rule) to be designed, and $y_i \in \Re^m$ is the output. The term $\Delta\mathcal{F}_i$ denotes the unknown agent dynamics.

Under this multiagent model, the distributed detection and control techniques to be developed will be rigorous, theoretically sound and practically feasible. Indeed, many practical multiagent systems, such as UAVs, are well described using the dynamical model in (1).

The (desired or emergent) group behaviors of multiagents are generated through modeling or design of local interaction rules based on sensing/communication-enabled local information exchange among agents. The interaction topology or sensing/communication structure can be represented by a digraph $\{\mathcal{V}, \mathcal{E}(t)\}$, where $\{\mathcal{V}\}$ denotes the set of N nodes and $\{\mathcal{E}\}$ is the set of directed edges. Accordingly, the local information flow among agents can be embedded into the following $N \times N$ binary sensing/communication matrix

$$S^n(t) = \left[S^n_{ij}(t)\right], \quad S^n_{ii} = 1, \tag{2}$$

where $S^n_{ij} = 1$ if $\{j \to i\} \in \mathcal{E}(t)$, and $S^n_{ij} = 0$ if otherwise.

To this end, the key research problems are (i) to analyze the overall behavior of the dynamic networked multiagent system in (1) through modeling and design of different local interaction rules v_i ; (ii) to analyze the unexpected/emergent behaviors of the dynamic networked multiagent system in (1) by conducting the estimation of interaction topologies $S^n(t)$ and the corresponding redesign of cooperative interaction control laws $v_i(t)$. Figure 1 illustrates the overall project architecture for addressing those research problems.

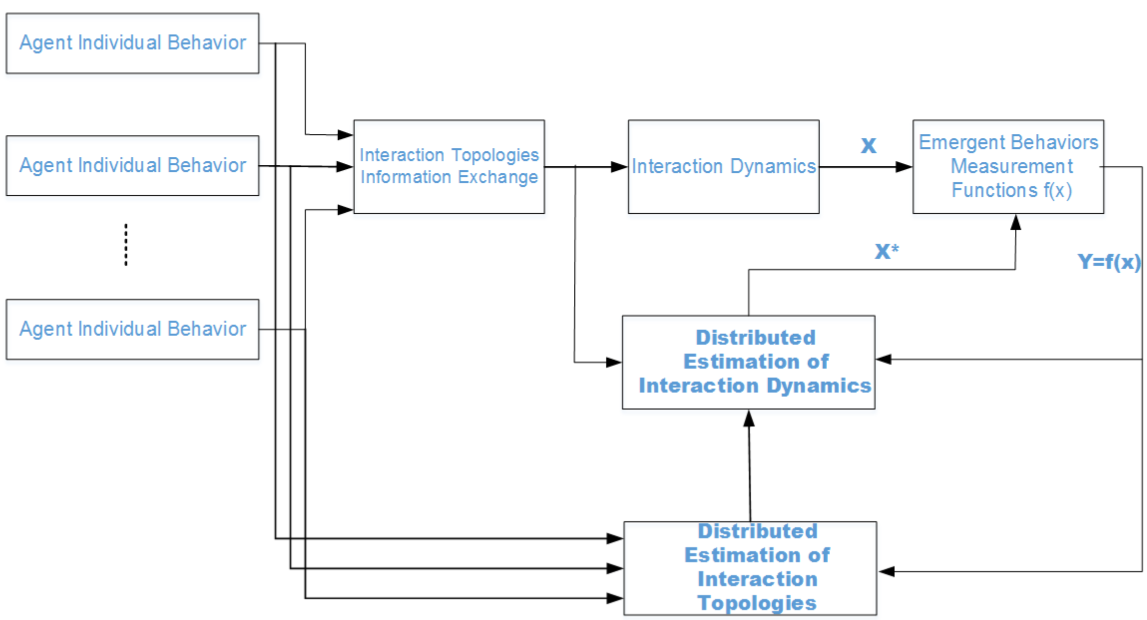

Figure 1: Project Architecture

In this project, by using approaches from systems and control theory, optimization and estimation, and adaptive learning and controls, the research on detection and control of emergent behaviors in multiagent systems was performed based on analysis and design of interaction dynamics. Specifically, we followed the following procedures to carry out research tasks.

- Perform model-based design, estimation, and control of multiagent interaction dynamics in the presence of uncertainties
- Design local algorithms for estimation of multiagent interaction topologies and emergent behavior detection
- Conduct simulation and experimental validation using AgentFly, UAV, and UGV platforms for multiagent emergent behavior control algorithms

In what follows, we report the obtained major research outcomes on distributed estimation algorithms, distributed optimization and control algorithms, and simulation and experiments in details.

4.0 RESULTS AND DISCUSSION: DISTRIBUTED AND RESILIENT ESTIMATION ALGORITHMS

The emergent behaviors in multiagent systems depend on the local interactions among agents [18, 7, 22, 11, 10, 1, 12]. We have studied the fundamental problems related to interaction topologies and interaction dynamics in multiagent systems, and obtained several results on detection of feature indicators for global behaviors, estimation of interaction topologies, and estimation of agents states using a limited number of sensors.

4.1 Distributed estimation of collective behaviors in multiagent systems

We proposed a distributed estimation algorithm for the detection of collective behaviors in multiagent systems [J3]. We assume that the collective behaviors of multiagent systems can be characterized by certain global characteristic signals such as network moments which may include centroid of the whole group, group polarization, group momentum, and so on. The accurate estimation of those global signals will enable us to detect the collective behaviors in multiagent systems. Apparently, in order to compute those global signals, it would be easier to use a central computer to collect and process state information of all agents in the group. However, such a method is not realistic if the number of agents is large. We provide a distributed solution for the estimation of those global time-varying signals, which is motivated by the distributed consensus algorithms [12]. Specifically, we design a local distributed estimator for each agent, and through information exchange with neighbors in its communication range, all the distributed estimators will reach consensus about the estimation of the group signals. To this end, by monitoring the estimate of any individual agent, we are be able to predict the group collective behaviors. The proposed design relies on the communication network among agents. By imposing information transmission whenever necessary, an event-triggering mechanism is adopted to further reduce the communication load. The convergence of the proposed algorithm is rigorously analyzed. A case study is given to study the collective behaviors of multiagent systems with unicycle models. Simulation examples verified the proposed method for collective behavior detection. Compared with the results such as in [16, 3], the proposed algorithm handles the estimation of time-varying signals, and the convergence is rigorously proved. In addition, an event-triggering strategy is proposed for ease of implementation of the proposed algorithm.

Specifically, the distributed estimation problem is formulated as follows. Consider a group of N agents, and agent i has the measurement of a time-varying signal $r_i(t)$. The objective is to find the value of average of all $r_i(t), i = 1, \cdots, N$, that is, $r(t) = \frac{\sum_{i=1}^{N} r_i(t)}{N}$. We designed a distributed estimator to find $r(t)$. Let $x_i(t)$ be the estimate of $r(t)$ by agent i, and the distributed estimation algorithm is given by

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} c_{ij}(x_j(t) - x_i(t)) + \dot{r}_i(t) \tag{3}$$

where $c_{ij} > 0$ if agent i can receive information from agent j. It follows from (3) that the overall closed-loop system dynamics are

$$\dot{x}(t) = -Lx(t) + \dot{r}(t) \tag{4}$$

where $x = [x_1, \cdots, x_n]^T$, $\dot{r} = [\dot{r}_1, \cdots, \dot{r}_n]^T$, and

$$L = \begin{bmatrix} \sum_{j \neq 1} c_{1j} & -c_{12} & \cdots & -c_{1n} \\ -c_{21} & \sum_{j \neq 2} c_{2j} & \cdots & -c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -c_{n1} & -c_{n2} & \cdots & \sum_{j \neq n} c_{nj} \end{bmatrix}$$

It is apparent that for the directed communication graph, the Laplacian matrix L is not symmetrical. If L is strongly connected, we have that corresponding to the eigenvalue $\lambda_1 = 0$, the right eigenvector is $\mathbf{1}$, and the left eigenvector is $w_1 = [p_1, p_2, \cdots, p_N]^T$ with $p_i > 0, \forall i$. In what follows, we first show that the algorithm (4) renders the weighted average of time-varying signals. It follows from (4) that

$$w_1^T \dot{x} = -w_1^T Lx + w_1^T \dot{r} = w_1^T \dot{r} \tag{5}$$

Thus, if the initial conditions satisfy $w_1^T x(0) = w_1^T r(0)$, then

$$w_1^T x(t) = w_1^T r(t) = \sum_{i=1}^{N} p_i r_i(t)$$

and if consensus is reached, we have

$$\lim_{t \to \infty} x_i(t) = \frac{\sum_{i=1}^{N} p_i r_i(t)}{\sum_{i=1}^{N} p_i} \tag{6}$$

The Proposed Distributed Estimation Algorithm: The algorithm in (4) leads to the estimation of the weighted average of signals $r_i(t)$. in order to obtain the estimation of the average of signals $r_i(t)$, we may consider to use the following modified distributed estimator

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} c_{ij}(x_j - x_i) + \frac{\dot{r}_i(t)}{Np_i} \tag{7}$$

It then follows

$$w_1^T \dot{x} = -w_1^T Lx + w_1^T \begin{bmatrix} \frac{\dot{r}_1(t)}{Np_1} \\ \vdots \\ \frac{\dot{r}_N(t)}{Np_N} \end{bmatrix} = \sum_{i=1}^{N} p_1 \frac{\dot{r}_i(t)}{Np_i} = \frac{\dot{r}_i(t)}{N}$$

To this end, we can see that if the consensus is reached, then

$$\lim_{t \to \infty} x_i(t) = \frac{\dot{r}_i(t)}{N} \tag{8}$$

However, in the algorithm (7), p_i is needed, which requires the information on communication network topology and in general agent i may not know it. To solve this problem, we propose the distributed estimation algorithm based on the estimation of left eigenvector. Let $\hat{w}^i = [\hat{p}_1^i, \hat{p}_2^i, \cdots, \hat{p}_N^i]^T$ be the estimate of w_i by agent i, and

$$\dot{\hat{w}}^i = \sum_{j \in \mathcal{N}_i} c_{ij}(\hat{w}^j - \hat{w}_i) \tag{9}$$

with the initial value $\hat{w}^i(0) = [0, \cdots, 1, \cdots, 0]^T$. To this end, the proposed distributed estimation algorithm is given by

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} c_{ij}(x_j - x_i) + \frac{\dot{r}_i(t)}{N\hat{p}_i} \tag{10}$$

$$\dot{\hat{p}}_i^i = \sum_{j \in \mathcal{N}_i} c_{ij}(\hat{p}_i^j - \hat{p}_i^i) \tag{11}$$

Simulation: The proposed algorithms (10) and (11) were simulated to estimate the average of time-varying signals $r_1(t) = 1 + e^{-0.2t}, r_2(t) = 5 + e^{-0.3t}, r_3(t) = 3 + e^{-0.4t}$. We assume that the communication topology among three agents renders to the following Laplacian matrix

$$L = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

and its left eigenvector is

$$w_1 = [0.2500, 0.5000, 0.2500]^T$$

Simulation results are shown in figures 2 and 3. It can be seen that all agents converge to the true average of signals.

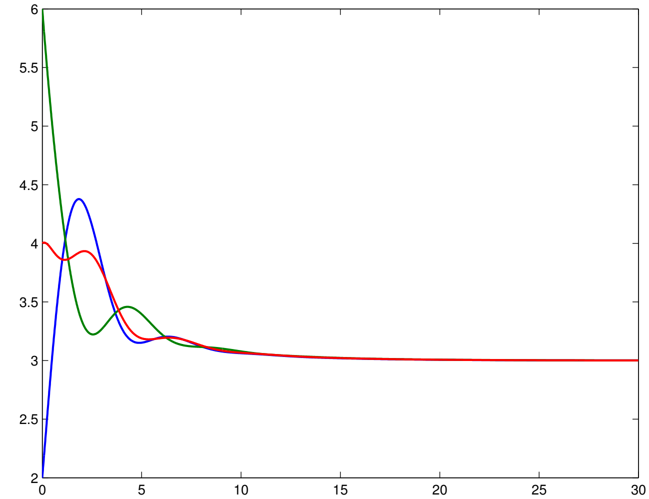

Figure 2: Estimates of the signal average by three agents

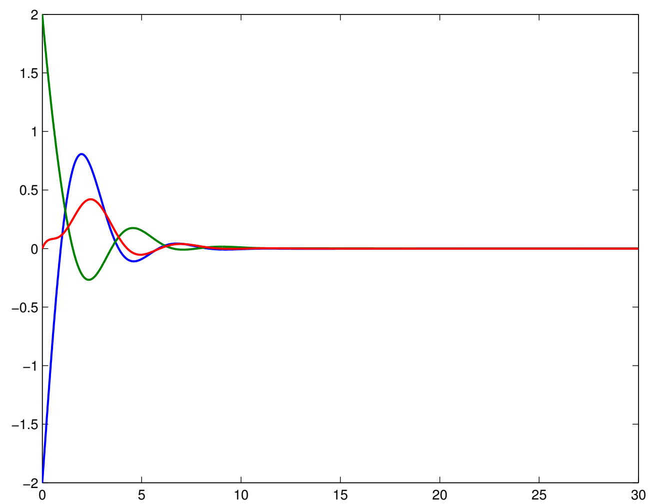

Figure 3: Estimation errors

The main results are reported and proved in **Theorem 1** [J-3]. Effects of communication jamming was also studied with distributed estimation of global features.

The published work in the above area is listed below.

[J-3] J. Wang, I.S. Ahn, Y. Lu, T. Yang, and G. Staskevich, "A Distributed Estimation Algorithm for Collective Behaviors in Multiagent Systems with Applications to Unicycle Agents", *International Journal of Control, Automation and Systems*, Springer, vol.15, no.6, pp. 2829-2839, 2017.

[C-11] Z. Fu, T. Yang, and J. Wang, Effects of Jamming on a Multi-agent Flocking Model with Distributed Estimation of Global Features, *SAI Intelligent Systems Conference 2016*, San Francisco, CA, Dec 6-7, 2016.

[C-12] J. Wang, I. S. Ahn, Y. Lu, and T. Yang, A distributed detection algorithm for collective behaviors in multiagent systems, *the 12th World Congress on Intelligent Control and Automation*, Guilin, China, June 12-15, 2016.

[C-13] T. Yang, Z. Fu, and J. Wang, Application of an even-triggered distributed estimation algorithm in a simple multiagent flocking model, *2016 IEEE SoutheastCon*, Norfolk VA, March 30-Apr 3, 2016.

[C-14] T. A. Khan and J. Wang, "On formalization of emergent behaviors in multiagent systems with limited interactions", *2016 IEEE International Conference on Electro Information Technology (EIT)*, Grand Forks, ND, May 19-21, 2016.

4.2 Distributed estimation with limited and unknown communications

The emergent behaviors of a large scale multiagent systems such as power grids can be monitored using wireless sensor networks (WSNs) with a limited number of low cost sensor nodes. One of the key issues in the application of WSNs is how the measurements by individual sensor components in the network can be effectively processed and utilized, the so-called sensor fusion problem. The common way of addressing this problem is to make individual sensors communicate their measurement data back to a central computer, and accordingly the combined information can be analyzed. However, this method may not be feasible in the situation with a large number of sensor nodes due to the inherent limitations on communication ranges, power supplies, memory, and computation power within sensor components.

To solve this challenge, we propose a new distributed least-squares estimation algorithm to detect the behaviors of the multiagent systems. We assume that the sensor network satisfies a general kind of network observability condition, that is, each sensor can take the measurements of a limited number of agents but the complete multiagent systems are covered under the union of all sensors in the network. In addition, we consider the limited communication among sensors and assume the communication topology among sensors is strongly connected. Each sensor is endowed with a distributed least-squares estimator. Through local information exchange with its communication neighbors, the estimation consensus can be reached for all sensors, and the state vector of all agents can be recovered by any individual sensors. Compared with the existing results such as those in [8, 5, 13, 4, 9], the proposed distributed least-squares algorithms can handle the directed communication network by explicitly estimating the left eigenvector corresponding to the largest eigenvalue of the system matrix. In addition, by introducing an elegantly structured observation matrix for each sensor, the possible singularity problem can be avoided even the number of sensors are significantly smaller than that of agents. The convergence of the proposed algorithm is analyzed, and simulation results further illustrate its effectiveness.

4.2.1 Distributed Least-Square Algorithm

In the proposed distributed least-square algorithm, we assume that each agent has a measurement model of the form

$$y_i(k) = H_i(k)x + v_i(k)$$

where $x = [x_1, x_2, \cdots, x_N]^T$ is an unknown vector representing the states of a number of N targets. The algorithm is summarized below in **Algorithm 1**, and details and simulation results can be found in [J-2].

4.2.2 Distributed Kalman Filtering Algorithm

The proposed distributed sensor fusion algorithm was further extended to multiagents with possibly unknown dynamics. Accordingly, a new distributed Kalman filtering algorithm was reported in [C-4]. We consider the detection of emergent behaviors of a group of n moving agents with the following dynamics

$$x_i(k+1) = \phi_i x_i(k) + \gamma_i w_i(k) \tag{12}$$

where $i = 1, \cdots, n$, $x_i \in \Re^{n_i}$ and ϕ_i is a $n_i \times n_i$ system matrix, γ_i is an $n_i \times m_i$ matrix, and $w_i \in \Re^{m_i}$ is the process noise with zero mean and diagonal, positive-definite covariance matrix $R_{w_i}(k)$. Assume that there are L sensors (detectors) which are employed to monitor the behaviors of all agents, and each sensor can only monitor the agents in its sensing range. Assume also $L \ll n$ (that is, the number of sensors is far more less than that of agents) and each sensor has a measurement model of the form

$$y_i(k) = H_i(k)X + v_i(k) \tag{13}$$

where $k = 0, 1, \cdots$ is an integer, y_i is a $p_i \times 1$ measurement vector, $X = [x_1^T, x_2^T, \cdots, x_n^T]^T$ is an $\sum_i n_i \times 1$ unknown vector representing the overall states of all agents, v_i is a $p_i \times 1$ white measurement noise vector with zero mean and diagonal, positive-definite covariance matrix $R_i(k)$, $H_i \in \Re^{p_i \times q}$ is the matrix relating the measurements to the unknowns, and $p = \sum_{i=1}^{N} p_i \geq n$. The dimension parameter p_i for sensor i depends on the number of agents in its range.

Algorithm 1 Distributed Algorithm

1: Initialization: $\hat{w}_i(0)$, $\bar{\mathcal{H}}_i(0) = H_i^T(0)R_i^{-1}(0)H_i(0)$, $\bar{\mathcal{Y}}_i(0) = H_i^T(0)R_i^{-1}(0)y_i(0)$, $\tilde{\mathcal{H}}_i(0) = 0$, $\tilde{\mathcal{Y}}_i(0) = 0$, $P_i(0) = (\hat{\mathcal{H}}_i(0))^{-1}$, $\hat{\mathcal{X}}_i(0) = P_i(0)\hat{\mathcal{Y}}_i(0)$, where $\hat{\mathcal{X}}_i$ is the estimate of x by sensor i.

2: **while** with new samples at time instant $k \geq 1$ **do**

3: Update $\hat{w}_i(k)$ using

$$\hat{w}_i(k+1) = \hat{w}_i(k) + \frac{1}{1+d_i} \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{w}_j(k) - \hat{w}_i(k))$$

4: Update $\hat{\mathcal{H}}_i(k)$ using

$$\begin{aligned} \tilde{\mathcal{H}}_i(k+1) &= \tilde{\mathcal{H}}_i(k) + \frac{1}{1+d_i} \sum_{j \in \mathcal{N}_i} (\tilde{\mathcal{H}}_j(k) - \tilde{\mathcal{H}}_i(k) + \bar{\mathcal{H}}_j(k) - \bar{\mathcal{H}}_i(k)) \\ \hat{\mathcal{H}}_i(k) &= \tilde{\mathcal{H}}_i(k) + \bar{\mathcal{H}}_i(k) \end{aligned}$$

5: Update $\hat{\mathcal{Y}}_i(k)$ using

$$\begin{aligned} \tilde{\mathcal{Y}}_i(k+1) &= \tilde{\mathcal{Y}}_i(k) + \frac{1}{1+d_i} \sum_{j \in \mathcal{N}_i} (\tilde{\mathcal{Y}}_j(k) - \tilde{\mathcal{Y}}_i(k) + \bar{\mathcal{Y}}_j(k) - \bar{\mathcal{Y}}_i(k)) \\ \hat{\mathcal{Y}}_i(k) &= \tilde{\mathcal{Y}}_i(k) + \bar{\mathcal{Y}}_i(k) \end{aligned}$$

6: Compute $P_i(k)$ using

$$P_i(k) = (P_i(k-1)^{-1} + \hat{\mathcal{H}}_i(k))^{-1}$$

7: Compute $\hat{\mathcal{X}}_i(k)$ using

$$\hat{\mathcal{X}}_i(k) = \hat{\mathcal{X}}_i(k-1) + P_i(k)(\hat{\mathcal{Y}}_i(k) - \hat{\mathcal{H}}_i(k)\hat{\mathcal{X}}_i(k-1))$$

8: **end while**

Assume that each sensor can communicate wirelessly with other sensors in its communication range r_c. We use a matrix to capture the communication topology within the sensor network. Let the adjacency matrix be

$$\mathcal{A}(k) = \begin{bmatrix} 0 & a_{12}(k) & \cdots & a_{1L}(k) \\ a_{21}(k) & 0 & \cdots & a_{2L}(k) \\ \vdots & \vdots & \ddots & \vdots \\ a_{L1}(k) & a_{L2}(k) & \cdots & 0 \end{bmatrix} \tag{14}$$

where $a_{ij}(k) > 0$ denotes that sensor j can transmit data to sensor i, otherwise $a_{ij}(k) = 0$.

The objective is to design a distributed estimation algorithm so that the estimation of all agents' states X can be done by individual sensors through information exchange.

Centralized Kalman Filtering Design: If measurements from all sensors are available, a centralized Kalman filter can be designed as follows. It follows from (12) and (13) that the overall agent dynamics and measurement model are

$$\begin{aligned} X(k+1) &= \Phi X(k) + \Gamma W(k) && (15) \\ Y(k) &= H(k)X(k) + V(k) && (16) \end{aligned}$$

where $Y(k) = [y_1^T(k), y_2^T(k), \cdots, y_L^T(k)]^T \in \Re^{p\times 1}$, $W(k) = [w_1^T(k), w_2^T(k), \cdots, w_n^T(k)]^T \in \Re^{\sum_i m_i \times 1}$, $V(k) = [v_1^T(k), v_2^T(k), \cdots, v_L^T(k)]^T \in \Re^{p\times 1}$, and

$$\Phi = \begin{bmatrix} \phi_1 & & & \\ & \phi_2 & & \\ & & \ddots & \\ & & & \phi_n \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \gamma_1 & & & \\ & \gamma_2 & & \\ & & \ddots & \\ & & & \gamma_n \end{bmatrix}, \quad H(k) = \begin{bmatrix} H_1(k) \\ H_2(k) \\ \vdots \\ H_L(k) \end{bmatrix}$$

For the overall dynamical systems (15) and (16), the Kalman filtering algorithm can be designed as follows

- Measurement update (at the measurement time $k = 1, 2, \cdots,$)

$$\begin{aligned} \hat{X}(k) &= \bar{X}(k) + P(k)\left[\sum_{i=1}^{L} H_i^T(k)R_i^{-1}(k)y_i(k) - \sum_{i=1}^{L} H_i^T(k)R_i^{-1}(k)H_i(k)\bar{x}(k)\right] && (17) \\ P(k) &= \left[M(k)^{-1} + \sum_{i=1}^{L} H_i^T(k)R_i^{-1}(k)H_i(k)\right]^{-1} && (18) \end{aligned}$$

- Time update (between the measurements)

$$\begin{aligned} \bar{X}(k) &= \Phi\hat{X}(k-1) && (19) \\ M(k) &= \Phi P(k-1)\Phi^T + \Gamma R_w \Gamma^T && (20) \end{aligned}$$

where $\bar{X}(k)$ is the estimate at time instant k given the data up through $k-1$, $\hat{X}(k)$ is the state estimate at time instant k given the data up through k, $M(k)$ is the error covariance of the state estimate $\bar{X}(k)$ before the measurement at k, $P(k)$ is the error covariant of the state estimate $\hat{X}(k)$ after the measurement at k, and the initial conditions are given by

$$\hat{x}(0) = P(0)\left(\sum_{i=1}^{L} H_i^T(0)R_i^{-1}(0)y_i(0)\right)$$

and $P(0) = \left(\sum_{i=1}^{L} H_i^T(0)R_i^{-1}(0)H_i(0)\right)^{-1}$.

In the implementation of (17) to (20), the computation of terms

$$\sum_{i=1}^{L} H_i^T(k)R_i^{-1}(k)H_i(k) \text{ and } \sum_{i=1}^{L} H_i^T(k)R_i^{-1}(k)y_i(k)$$

requires all sensors to send their measurements matrices $H_i(k)$, covariance matrices $R_i(k)$, and raw measurements $y_i(k)$ to a fusion center, which is not realistic due to the issues with scalability, fault tolerance, and communication constraints. In what follows, we present a distributed Kalman filtering algorithm.

Distributed Kalman Filtering Algorithm: Let $\hat{X}_i$ be the estimate of X by sensor i, $\hat{\mathcal{H}}_i(k)$ be the estimate of $\sum_{i=1}^{L} H_i^T(k)R_i^{-1}(k)H_i(k)$ and $\hat{\mathcal{Y}}_i(k)$ be the estimate of $\sum_{i=1}^{L} H_i^T(k)R_i^{-1}(k)y_i(k)$ by sensor i, respectively. The proposed distributed Kalman filtering algorithm is given by

- Measurement update (at the measurement time $k = 1, 2, \cdots,$)

$$\begin{aligned}
\hat{X}_i(k) &= \bar{X}_i(k) + P_i(k)\left[\hat{\mathcal{Y}}_i(k) - \hat{\mathcal{H}}_i(k)\bar{x}(k)\right] && (21)\\
P_i(k) &= \left[M_i(k)^{-1} + \hat{\mathcal{H}}_i(k)\right]^{-1} && (22)\\
\hat{\mathcal{H}}_i(k+1) &= \hat{\mathcal{H}}_i(k) + \frac{1}{1+d_i}\sum_{j\in\mathcal{N}_i} a_{ij}(\hat{\mathcal{H}}_j(k) - \hat{\mathcal{H}}_i(k)) \\
&\quad +\beta_i[H_i^T(k+1)R_i^{-1}(k+1)H_i(k+1) - H_i^T(k)R_i^{-1}(k)H_i(k)], && (23)\\
\hat{\mathcal{Y}}_i(k+1) &= \hat{\mathcal{Y}}_i(k) + \frac{1}{1+d_i}\sum_{j\in\mathcal{N}_i} a_{ij}(\hat{\mathcal{Y}}_j(k) - \hat{\mathcal{Y}}_i(k)) \\
&\quad +\beta_i[H_i^T(k+1)R_i^{-1}(k+1)y_i(k+1) - H_i^T(k)R_i^{-1}(k)y_i(k)], && (24)
\end{aligned}$$

- Time update (between the measurements)

$$\begin{aligned}
\bar{X}_i(k) &= \hat{\Phi}_i(k-1)\hat{X}_i(k-1) && (25)\\
M_i(k) &= \hat{\Phi}_i(k-1)P_i(k-1)\hat{\Phi}_i^T(k-1) && (26)\\
\hat{\Phi}_i(k) &= \hat{\Phi}_i(k-1) + \frac{1}{1+d_i}\sum_{j\in\mathcal{N}_i}(\hat{\Phi}_j(k-1) - \hat{\Phi}_i(k-1)) && (27)
\end{aligned}$$

where $\hat{\Phi}_i(k)$ is the estimate of Φ by agent i with the initial conditions $\hat{\mathcal{H}}_i(0) = \beta_i H_i^T(0) R_i^{-1}(0) H_i(0)$, $\hat{\mathcal{Y}}_i(0) = \beta_i H_i^T(0) R_i^{-1}(0) y_i(0)$,

$$\hat{\Phi}_i(0) = \beta_i \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & \phi_i & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix}$$

and $\beta_i = \frac{1}{\hat{w}_{ii}}$ with $\hat{w}_{ii}$ being updated by

$$\hat{w}_i(k+1) = \hat{w}_i(k) + \frac{1}{1+d_i} \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{w}_j(k) - \hat{w}_i(k)) \tag{28}$$

where $\hat{w}_i = [\hat{w}_{i1}, \hat{w}_{i2}, \cdots, \hat{w}_{iN}] \in \Re^N$ is an estimation vector generated by sensor i, $\hat{w}_i(0) = [0, \cdots, 1, \cdots, 0]^T$ with its ith element being 1, $d_i = \sum_{j \neq i} a_{ij}$, and $\mathcal{N}_i \triangleq \{j | a_{ij} > 0\}$ defines the neighboring set for agent i.

We also developed a strategy to handle the unpredictable changes of communication topologies. In such a case, the estimation of the corresponding left eigenvector under the new topology has to be redone to capture the unexpected change of link connectivity. To do so, we let the estimators in (28) periodically reset their initial values to $\hat{w}_i(0) = [0, \cdots, 1, \cdots, 0]^T$. In other words, the following eigenvector estimator will be used

$$\begin{aligned} \hat{w}_i(k+1) &= \hat{w}_i(k) + \frac{1}{1+d_i} \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{w}_j(k) - \hat{w}_i(k)), \text{for } k \in [\tau T, (\tau+1)T) \qquad (29) \\ \hat{w}_i(\tau T) &= [0, \cdots, 1, \cdots, 0]^T \qquad (30) \end{aligned}$$

where the integer $\tau = 0, 1, \cdots$, and the integer T is the period of resetting. Below is an example to illustrate the estimate of communication topology changes.

Example 1 *Assume that four agents switch their communication topologies according to the following graphs. The corresponding adjacency matrices are*

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

and the system matrices are

$$F_1 = \begin{bmatrix} 0.5000 & 0 & 0.5000 & 0 \\ 0.5000 & 0.5000 & 0 & 0 \\ 0 & 0.3333 & 0.3333 & 0.3333 \\ 0.5000 & 0 & 0 & 0.5000 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 0.5000 & 0.5000 & 0 & 0 \\ 0 & 0.5000 & 0.5000 & 0 \\ 0 & 0 & 0.5000 & 0.5000 \\ 0.5000 & 0 & 0 & 0.5000 \end{bmatrix}$$

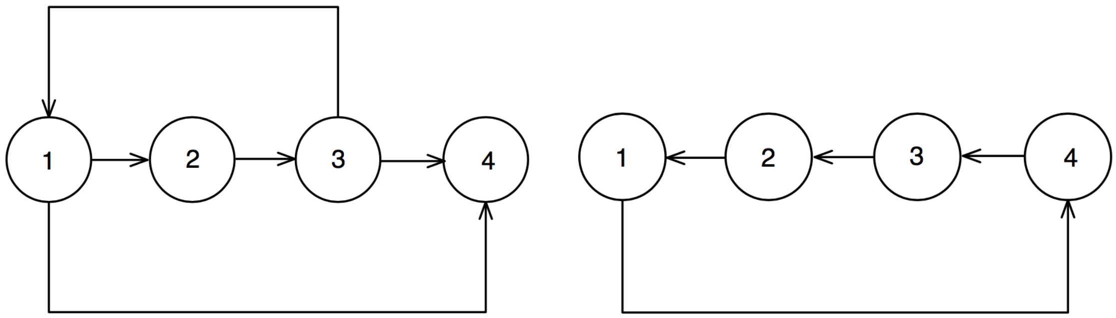

Figure 4: Communication topologies: (A_1: Left, A_2: Right)

The left eigenvectors to be estimated for F_1 and F_2 are

$$w_{1,F_1} = \begin{bmatrix} 0.3636 \\ 0.1818 \\ 0.2727 \\ 0.1818 \end{bmatrix}, \quad w_{1,F_2} = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}$$

In the simulation, for $k \in [0, 100)$, four agents assume communication topology A_1; for $k \in [100, 200)$, four agents assume communication topology A_2; and for $k \geq 200$, four agents assume communication topology A_1 again. The period for resetting initial values of estimators in (29) is $T = 50$. The estimation results are depicted in figures 5 to 8. It can be seen from Figure 5 that for

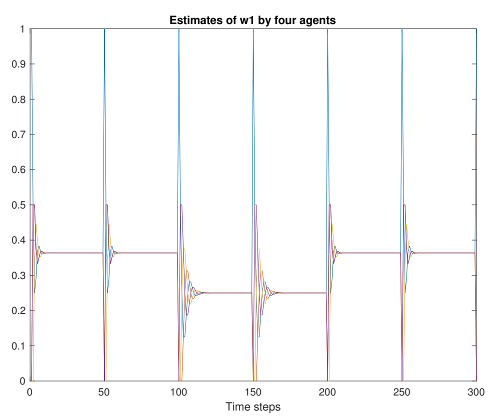

Figure 5: Estimates of w_{11} by four agents

$k \in [0, 100)$, *the estimates of w_{11} by four agents converge to 0.3636, which is the first component of w_{1,F_1}; for $k \in [100, 200)$, the estimates of w_{11} by four agents converge to 0.25, which is the first component of w_{1,F_2}; and $k \in [200, 300)$, the estimates of w_{11} by four agents converge to 0.3636, which is the first component of w_{1,F_1}. It should also note that every 50 steps, the initial values of estimators are reset no matter whether there is a change of communication topology or not. However this is necessary to capture communication changes sooner or later.*

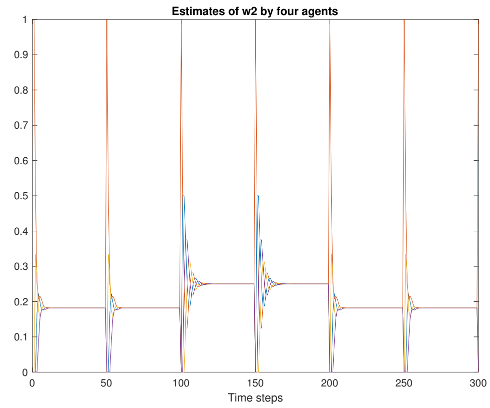

Figure 6: Estimates of w_{12} by four agents

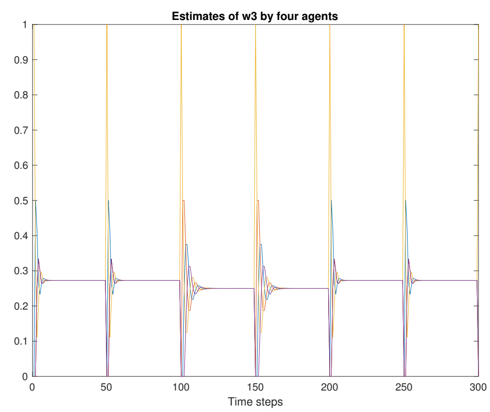

Figure 7: Estimates of w_{13} by four agents

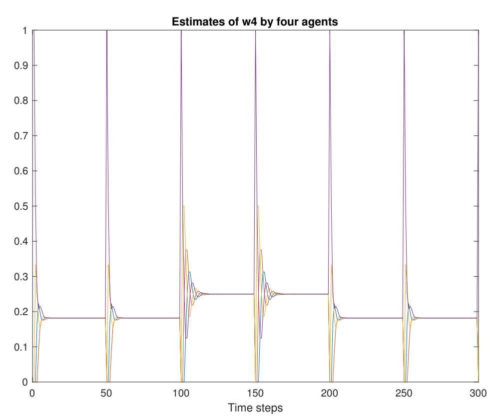

Figure 8: Estimates of w_{14} by four agents

The published work in the above area is listed below.

[J-2] J. Wang, I. S. Ahn, Y. Lu, T. Yang, and G. Staskevich, "A distributed least-squares algorithm in wireless sensor networks with limited and unknown communications", *International Journal of Handheld Computing Research*, vol.8, no.3, pp. 15-36, 2017. (DOI: 10.4018/IJHCR.2017070102)

[C-4] J. Wang, "Distributed estimation of moving targets with unknown dynamics ", *2018 IEEE Aerospace Conference*, Big Sky, MT, March 3-10, 2018.

[C-7] J. Wang, I. S. Ahn, Y. Lu, T. Yang, and G. Staskevich, "A distributed least-squares algorithm in wireless sensor networks with limited communication", *17th IEEE International Conference on Electro Information Technology (EIT)*, Lincoln, Nebraska, May 14-17, 2017.

[C-9] M. Imtiaz and J. Wang, "A multiagent reinforcement learning control approach to environment exploration", *2017 IEEE SouthEastCon*, Charlotte NC, Mar 30-Apr. 2, 2017.

[C-10] J. Wang, I. S. Ahn, Y. Lu, and G. Staskevich, A New Distributed Algorithm for Environmental Monitoring by Wireless Sensor Networks with Limited Communication, *2016 IEEE Sensors*, Orlando, FL, Oct 30 Nov 2, 2016.

4.3 Resilient detection of multiple targets

We have also studied the resilience and robustness of the proposed distributed estimation algorithm (**Algorithm 1**) in the presence of jamming, limited sensing/limited communication, unexpected agent failure, unexpected communication link dropout, and the situation with intermittent communications. We assume that there are M agents (detectors/sensors), which will be used to collaboratively detect the behaviors of N targets. The number of agents is much smaller than that of targets (i.e., $M \ll N$). Targets are assumed to be located in a 2D environment. Each agent has a limited sensing/communication range and can only detect a small group of targets in its sensing range. Agents maintain a strongly connected communication topology and each agent can communicate with its neighboring agents about their situation of target detection. Below are several scenarios we studied for the illustration of algorithm resilience.

Scenario #1: Limited sensing/Limited communication; measurement noises

As shown in figure 9, there are five agents (green circles) used for detection of 20 target (magenta circles). Targets are randomly placed in a region of dimension $[-10, 10]^2$. The sensing range of agent is $R = 5\sqrt{2}$. The communication topology of 5 agents is given in figure 10. The estimation noise is Gaussian noise with zero mean and variance 0.25.

Apparently, each agent can only detect a limited number of targets. By using the proposed distributed estimation algorithm, each agent will be able to estimate the positions of the total 20 targets. The performance of the algorithm is measured using the position root mean square error

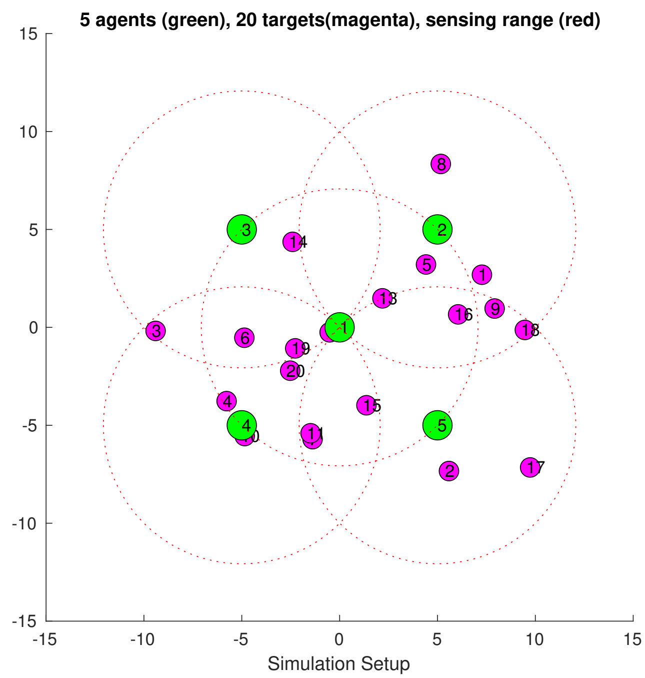

Figure 9: Simulation Setup

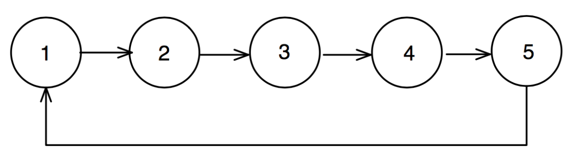

Figure 10: Communication Topology among Agents

value (RMSE) computed using the following formula.

$$RMSE(k) = \sum_{i=1}^{20} \sqrt{(\hat{x}_i(k) - x_i)^2 + (\hat{y}_i(k) - y_i)^2}$$

where (x_i, y_i) is the position of the ith target, $(\hat{x}_i(k), \hat{y}_i(k))$ is the position estimate of the ith target by agents at time step k. The RMSE values for all agents are depicted in figure 11, which converge to a small region of zero.

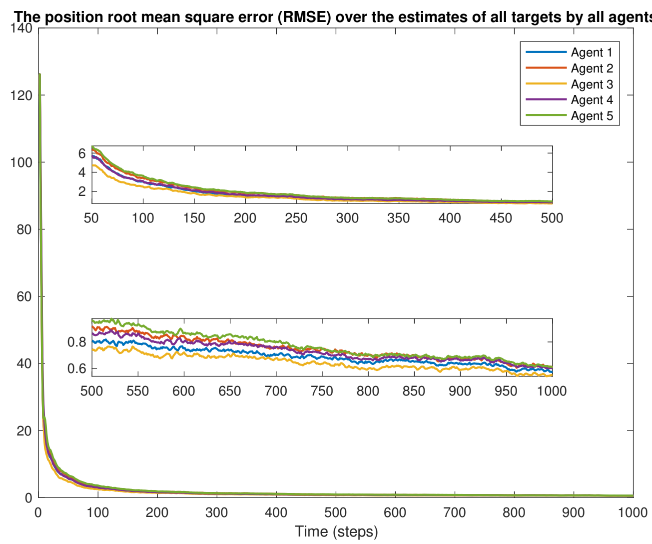

Figure 11: The Position RMSE Values for all targets by all agents

Figure 12 and 13 show the performance of estimation with a large measurement noise variance, that is, $\sigma = 5$. It can be seen the proposed algorithm still works as expected.

Scenario #2: Agent failure

In this case, we consider the scenario with blind agents or unexpected agent failure. We use the similar simulation setup as scenario 1, but assume that Agent 1 has failure and does not provide any measurements. But it can still communicate with other agents based on the topology in figure 10.

Figure 14 and 15 show the performance of estimation with a measurement noise variance 0.5. It can be seen the proposed algorithm still works as expected even there is failure with agent 1.

Scenario #3: Unexpected communication link dropout: switching topologies

In this case, we consider the scenario that the communication topologies among agents periodically switch between two patterns showing in figure 16. In other words, every 500 steps, the topology switches.

Figure 17 and 18 show the performance of estimation with a measurement noise variance 0.5. It

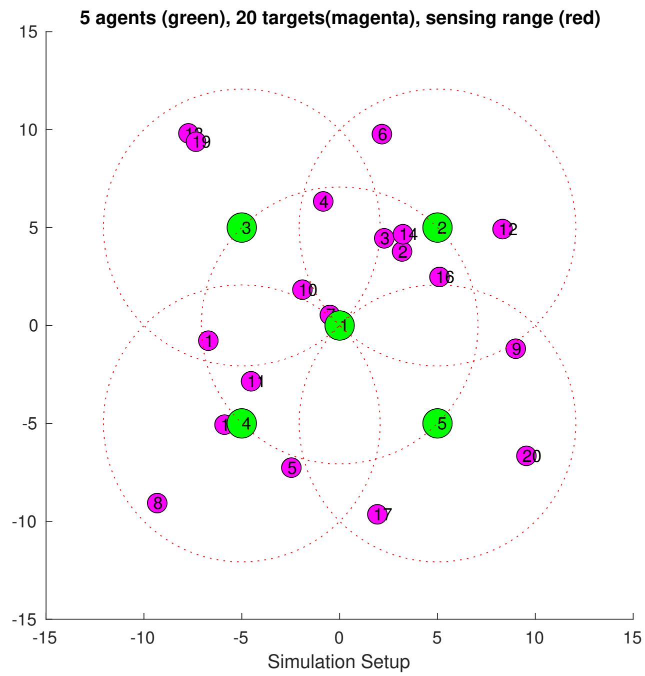

Figure 12: Simulation Setup (noise variable = 5)

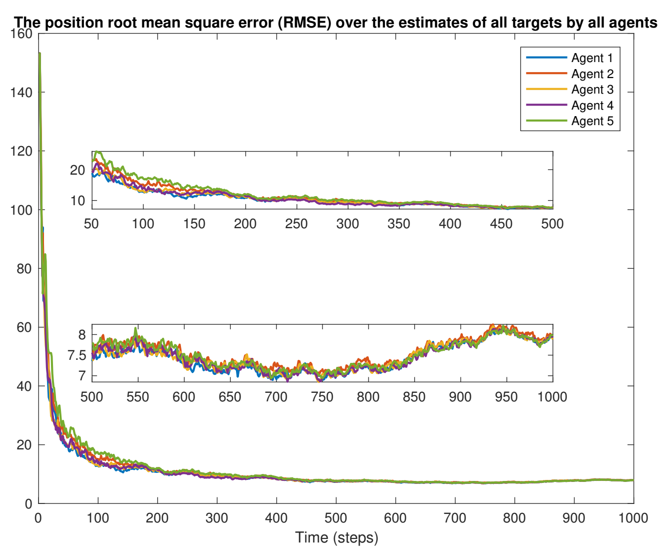

Figure 13: The Position RMSE Values for all targets by all agents

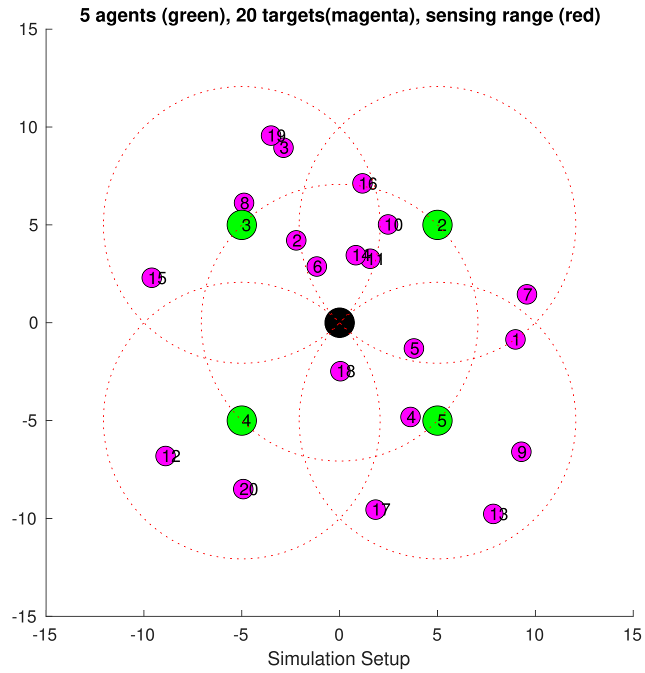

Figure 14: Simulation Setup (noise variable =0.5)

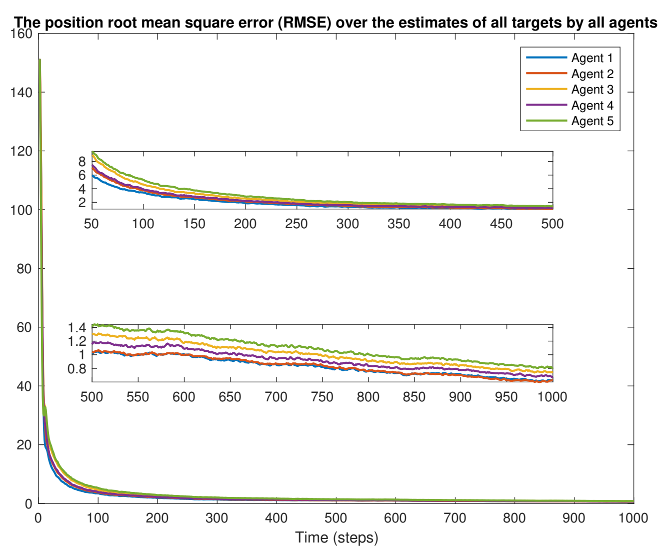

Figure 15: The Position RMSE Values for all targets by all agents

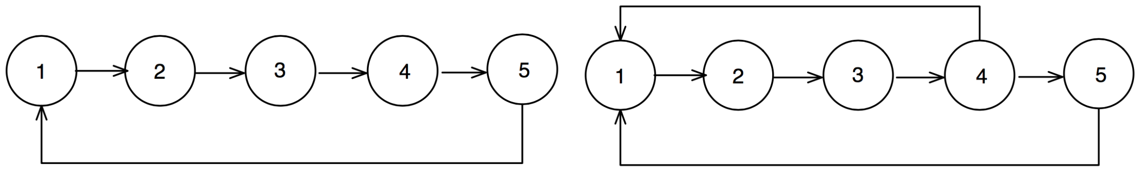

Figure 16: Communication Topologies

can be seen the proposed algorithm still works as expected under the switching communication topologies.

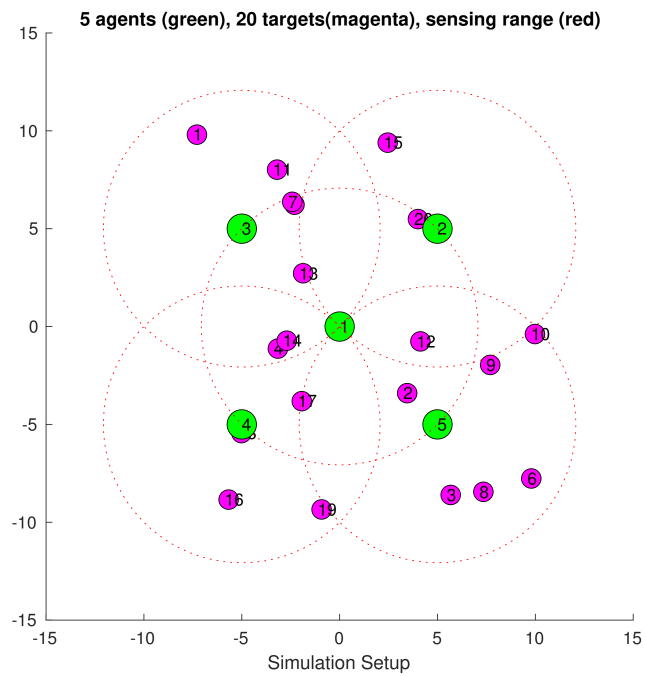

Figure 17: Simulation Setup (noise variable =0.5)

Scenario #4: Unexpected communication link dropout: with not strongly connected topologies

In this case, we assume the communication topologies switch between patterns showing in figure 19. It can be seen that the left hand one is not strongly connected.

Figure 20 and 21 show the performance of estimation with a measurement noise variance 0.5. It can be seen that the estimation error bounds are still in the acceptable range.

The published work in the above area is listed below.

[C-1] J. Wang, I. S. Ahn, Y. Lu, T. Yang, and G. Staskevich, “Resilient detection of multiple

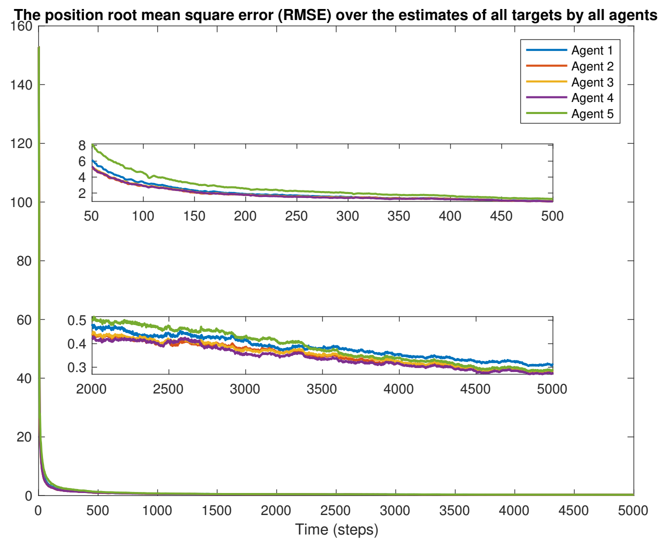

Figure 18: The Position RMSE Values for all targets by all agents

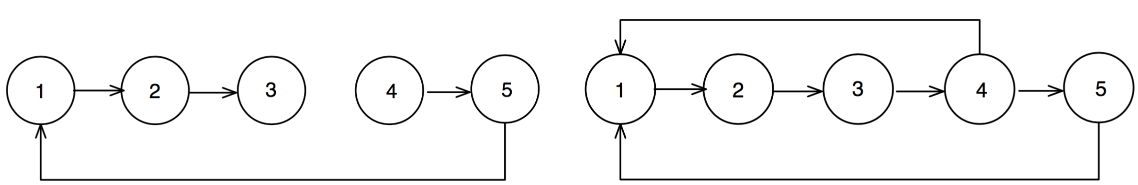

Figure 19: Communication Topologies

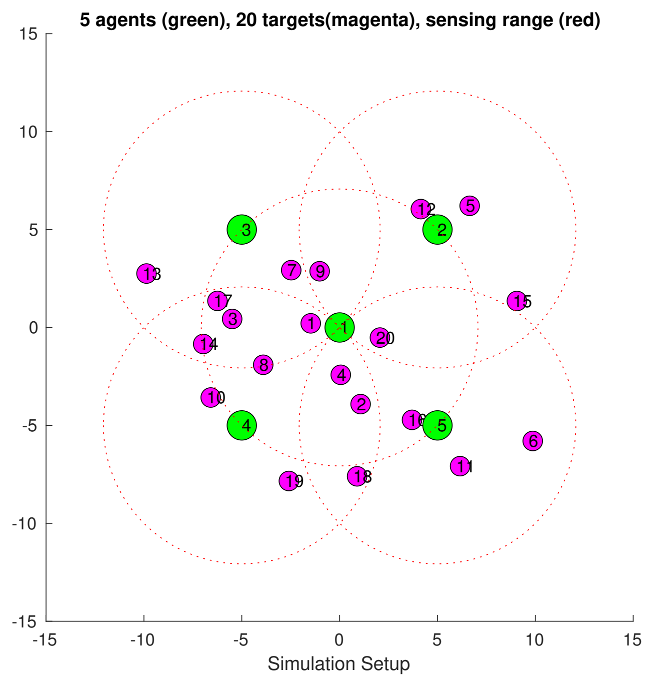

Figure 20: Simulation Setup (noise variable =0.5)

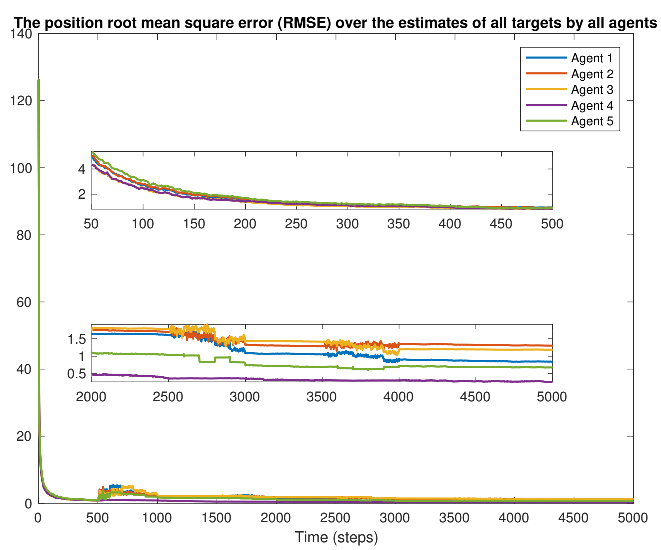

Figure 21: The Position RMSE Values for all targets by all agents

targets using a distributed algorithm with limited information sharing", *2018 SPIE Defense Conference*, Orlando, FL, April 16 – April 19, 2018.

[C-5] H. Liu, R. Cheng, T. Yang, and J. Wang, "Modeling and verifying the communication and control of a fleet of collaborative autonomous underwater vehicles," *43th Annual Conf. of the IEEE Industrial Electronics Society (IECON17)*, Beijing, Oct. 29-Nov.1, 2017.

[C-11] Z. Fu, T. Yang, and J. Wang, Effects of Jamming on a Multi-agent Flocking Model with Distributed Estimation of Global Features, *SAI Intelligent Systems Conference 2016*, San Francisco, CA, Dec 6-7, 2016.

5.0 RESULTS AND DISCUSSION: DISTRIBUTED OPTIMIZATION AND CONTROL

Distributed control of agent behaviors and networked optimization are of paramount importance in the study of multiagent system emergent behaviors. One of the fundamental issues is how to design local and distributed control to coordinate the individual agent's behavior such that the desired group behavior emerges. Following this line of research, we have developed adaptive cooperative control algorithms for multiagents with uncertainties and an approximated distributed gradient estimation algorithms for networked system optimization.

5.1 Distributed coordinated tracking of multiagents

We have studied the distributed coordinated tracking control for multiagent systems with model uncertainties. Both unknown model parameters and unknown system dynamics are considered. It is assumed that there exist parametric uncertainties and unknown dynamics with the informed agent as well, and only the state value of the informed agent can be accessed by a limited number of agents. With the utilization of neural network approximation and adaptive estimation, a new distributed adaptive tracking control is proposed to make all agents cooperatively follow the desired trajectory specified by the informed agent. The control design is first presented for the first-order multiagent systems, and then extension is made to the second-order multiagent systems using backstepping. A unique feature of the proposed control is that the unknown bounds of neural network approximation errors are also estimated online. Using Lyapunov stability theorem, it is rigorously proved that asymptotically cooperative tracking can be achieved under the assumption that the sensing/communication topology among agents is connected. Simulation results illustrated the effetiveness of the proposed control.

Specifically, consider a multiagent system defined by a set of scalar differential equations given below

$$\dot{x}_i = f_i(x_i) + u_i, \tag{31}$$

where $i \in \{1, \cdots, N\}$ is the index for agent i and there are N agents in the group, $x_i \in \Re$ is the state, $u_i \in \Re$ is the control to be designed, and $f_i(x_i)$ is a smooth function of its argument with $f(0) = 0$ representing the unknown system dynamics. We assume that there is an informed agent (leader) whose dynamics are described by $\dot{x}_0 = a_0 x_0 + r_0(t)$, where constant $a_0 < 0$, and $r_0(t)$ is a piecewise-continuous bounded function of time. We further assume that a_0 and $r_0(t)$ are completely unknown to all agents, and $x_0(t)$ may be sensed or communicated to some agents in the group. The objective is to design a control input $u_i(t), t \geq t_0$ based on the information exchange among agents so that all the signals in the multiagent system remain bounded and

$$\lim_{t\to\infty} |x_i(t) - x_0(t)| = 0, \quad \forall i. \tag{32}$$

Control u_i relies on the information exchange among agents, which can be described using the sensing/communication matrix defined below

$$S = [s_{ij}], \tag{33}$$

where $s_{ii} = k$ for all i with some constant $k > 0$, $s_{ij} = s_{ii} > 0$ if the ith agent can obtain the information from the jth agent, and $s_{ij} = 0$ if otherwise. We assume that all agents have equal sensing/communication capabilities, that is, $s_{ij} = s_{ji}$ and S is symmetric. Accordingly, the Laplacian matrix L induced by S is defined as

$$L = \text{diag}\left\{\sum_{j=1}^{n} s_{ij}\right\} - S. \tag{34}$$

We also assume that the informed agent state $x_0(t)$ is available to at least one agent through sensing/communication detection, and this is described by a diagonal matrix B given below

$$B = \text{diag}\{b_{i0}\}. \tag{35}$$

where $b_{i0} > 0$ means that agent i has the information $x_0(t)$.

Proposed Adaptive Neural Control for Multiagent Systems: To facilitate the design, suppose that $f_i(x_i)$ is parameterized using a linearly parameterized neural network as $f_i(x_i) = \psi_i^T(x_i)\theta_i + \epsilon_i$, and $r_0(t)$ is parameterized as $r_0(t) = \phi^T(t)w$, where basis functions $\psi_i(x_i) = [\psi_{i,1}, \cdots, \psi_{i,n_i}] \in \Re^{n_i}$, $\phi(t) = [\phi_1(t), \phi_2(t), \cdots, \phi_l(t)]^T \in \Re^l$ are known, and parameters $\theta_i = [\theta_{i,1}, \cdots, \theta_{i,n_i}]^T \in \Re^{n_i}$, $w = [w_1, w_2, \cdots, w_l]^T \in \Re^l$ are unknown constants, and ϵ_i is the neural network approximation error. Based on the universal approximation result for neural network, we assume that ϵ_i is bounded by an unknown constant δ_i, that is, $|\epsilon_i| \leq \delta_i$.

Let $\hat{\theta}_{i,j}$ be the estimate of $\theta_{i,j}$ for $j = 1, \cdots, n_i$, $\hat{a}_i$ be the estimate of a_0 by agent i, $\hat{w}_{ij}$ be the estimate of w_j by agent i, and $\hat{\delta}_i$ be the estimate of δ_i. Let $\hat{\theta}_i = [\hat{\theta}_{i,1}, \cdots, \hat{\theta}_{i,n_i}]^T$ and $\hat{w}_i = [\hat{w}_{i1}, \cdots, \hat{w}_{il}]^T$.

Define $e_i = \sum_{j\in\mathcal{N}_i} s_{ij}(x_i - x_j) + b_{i0}(x_i - x_0)$ with $\mathcal{N}_i = \{j | s_{ij} = 1\}$ denoting the neighboring set of agent i. The control input for agent i is chosen to be of the form

$$u_i = \hat{a}_i x_i - \psi_i^T(x_i)\hat{\theta}_i - \text{sgn}(e_i)\hat{\delta}_i + \phi^T(t)\hat{w}_i, \tag{36}$$

where $\mathrm{sgn}(x)$ is defined as $\mathrm{sgn}(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0. \end{cases}$

Define the tracking error $\tilde{X} = [\tilde{x}_1, \cdots, \tilde{x}_N]^T = [x_1 - x_0, \cdots, x_N - x_0]^T$, the parameter estimation errors $\tilde{a}_i = \hat{a}_i - a_0$, $\tilde{\theta}_i = \hat{\theta}_i - \theta_i = [\tilde{\theta}_{i,1}, \cdots, \tilde{\theta}_{i,n_i}]^T$, $\tilde{w}_i = \hat{w}_i - w_i = [\tilde{w}_{i1}, \cdots, \tilde{w}_{il}]^T$, and $\tilde{\delta}_i = \hat{\delta}_i - \delta_i$. It then follows from (31) and (36) that

$$\dot{\tilde{x}}_i = a_0\tilde{x}_i + \tilde{a}_i x_i - \psi_i^T\tilde{\theta}_i + \phi^T\tilde{w}_i + \epsilon_i - \mathrm{sgn}(e_i)\hat{\delta}_i, \tag{37}$$

and the overall N system error equation can be derived as

$$\dot{\tilde{X}} = a_0\tilde{X} + \mathcal{X}\tilde{a} - \Psi + \sum_{j=1}^{l} \Phi_j\tilde{w}_{*j} + \epsilon - \Delta \tag{38}$$

where $\tilde{a} = [\tilde{a}_1, \cdots, \tilde{a}_N]^T$, $\Psi = [\tilde{\theta}_1^T\psi_1, \cdots, \tilde{\theta}_N^T\psi_N]^T$, $\tilde{w}_{*j} = [\tilde{w}_{1j}, \cdots, \tilde{w}_{Nj}]^T$, $\mathcal{X} = \mathrm{diag}\,[x_1, \cdots, x_N]$, $\epsilon = [\epsilon_1, \cdots, \epsilon_N]^T$, $\Delta = [\mathrm{sgn}(e_1)\hat{\delta}_1, \cdots, \mathrm{sgn}(e_N)\hat{\delta}_N]^T$, and $\Phi_j = \mathrm{diag}\,[\phi_j(t), \cdots, \phi_j(t), \cdots, \phi_j(t)]$, $j = 1, \cdots, l$. The adaptive laws for $\hat{a}_i$, $\hat{\theta}_i$, $\hat{w}_i$, and $\hat{\delta}_i$ are given by

$$\dot{\hat{a}}_i = -\Gamma_{a_i}^{-1} x_i e_i, \tag{39}$$

$$\dot{\hat{\theta}}_i = \Gamma_{\theta_i}^{-1}\psi_i e_i, \tag{40}$$

$$\dot{\hat{w}}_{ij} = -\Gamma_{w_{ij}}^{-1}\phi_j e_i, \tag{41}$$

$$\dot{\hat{\delta}}_i = \Gamma_{\delta_i}^{-1}\mathrm{sgn}(e_i)e_i, \tag{42}$$

where $i = 1, \cdots, N$, $j = 1, \cdots, l$, $\Gamma_{a_i} > 0$, $\Gamma_{\theta_i} > 0$, $\Gamma_{w_{ij}} > 0$, and $\Gamma_{\delta_i} > 0$. The following theorem presents one of the main results of the paper.

Theorem 1 *Consider the multiagent system in (31). If the sensing/communication topology S is connected, and B has at least one entry being nonzero, then the distributed adaptive neural tracking control in (36) together with the adaptive laws in (39), (40), (41), and (42) guarantee the boundedness of all signals of the closed-loop system and achieve asymptotical tracking in the sense of (32).*

Extension to Second-Order Multiagent Systems: We further extend the proposed distributed adaptive neural control to the second-order multiagent systems. Consider the second-order multi-agent systems of the following form

$$\begin{cases} \dot{x}_{i1} = a_i x_{i1} + x_{i2} \\ \dot{x}_{i2} = f_i(x_{i1}, x_{i2}) + u_i, \qquad i = 1, \cdots, n \end{cases} \tag{43}$$

where a_i is an unknown constant, $f_i(x_{i1}, x_{i2})$ is an unknown smooth nonlinear function, $x_i = [x_{i1}, x_{i2}]^T \in \Re^2$ is the state of agent i, and $u_i \in \Re$ is the control input of agent i. The objective is to design a distributed adaptive control to make every agent i follow the informed agent, that is,

$$\lim_{t\to\infty} |x_{i1}(t) - x_0(t)| = 0, \quad \forall i. \tag{44}$$

The control design details can be found in our recent paper.

Simulation: A simulation example is given to illustrate the proposed adaptive neural tracking control for multiagent systems. Consider a group of 3 agents with the following dynamics

$$\begin{cases} \dot{x}_{11} = 2x_{11} + x_{12} \\ \dot{x}_{12} = 0.2e^{-x_{12}} + x_{11}\sin(x_{12}) + u_1 \end{cases}, \quad \begin{cases} \dot{x}_{21} = x_{21} + x_{22} \\ \dot{x}_{22} = 0.6\sin(x_{22})x_{21} + u_2 \end{cases}, \quad \begin{cases} \dot{x}_{31} = 5x_{31} + x_{32} \\ \dot{x}_{32} = 0.5(x_{31}^2 + x_{32}^2) + u_3 \end{cases}, \tag{45}$$

and the informed agent is $\dot{x}_0 = -x_0 + r(t)$, with $r(t) = 2\cos(0.5t)$. Systems in (45) are in the form of (43) with $a_1 = 2$, $f_1(x_{11}, x_{12}) = 0.2e^{-x_{12}} + x_{11}\sin(x_{12})$, $a_2 = 1$, $f_2(x_{21}, x_{22}) = 0.6\sin(x_{22})x_{21}$, and $a_3 = 5$, $f_3(x_{31}, x_{32}) = 0.5(x_{31}^2 + x_{32}^2)$. For the informed agent, we have $a_0 = -1$, $w = 2$, and $\phi(t) = \cos(0.5t)$. We assume that the sensing/communication matrix S is given by

$$S = \begin{bmatrix} k & k & 0 \\ k & k & k \\ 0 & k & k \end{bmatrix},$$

and agent 1 receives the information of $x_0(t)$, i.e., $B = \text{diag}\{k, 0, 0\}$, where constant $k > 0$. The parameters a_i, a_0, w and functions $f_i(x_{i1}, x_{i2})$ are assumed to be unknown in the simulation. We use RBF neural networks to approximate $f_i(x_{i1}, x_{i2})$, i.e., $f_i(x_{i1}, x_{i2}) = \psi_i^T(x_{i1}, x_{i2})\theta_i + \epsilon_i$, where $|\epsilon_i| \leq \delta_i$, and $\psi_i = [\psi_{i,1}, \cdots, \psi_{i,n_i}]^T$, with $\psi_{i,j}$ being chosen as the commonly used Gaussian functions, which have the form $\psi_{i,j} = e^{-(x_i - \mu_{i,j})^T(x_i - \mu_{i,j})/\eta_{i,j}^2}$, $j = 1, \cdots, n_i$, where $\mu_{i,j}$ is the center of the receptive field and η_i is the width of the Gaussian function. For the design of distributed adaptive neural control, let $\hat{a}_i, \hat{w}_i, \hat{a}_{0i}, \hat{\delta}_i, \hat{\theta}_i$ be the estimates of unknown parameters $a_0 - a_i$, w, a_0, δ_i, and θ_i, respectively.

The performance of the adaptive neural control relies on the selection of the centers and widths of RBF. For Gaussian RBF NNs, it was shown in that the centers can be arranged on a regular lattice on $\Re^n$ to uniformly approximate smooth functions. Accordingly, we select the widths and centers as: $\eta_{i,j} = 1, \forall i, j$, every neural network $\psi_i^T\theta_i$ contains 121 nodes, with center $\mu_{i,j}(j = 1, \cdots, 121)$ evenly spaced in $[-10, 10] \times [-10, 10]$. The following initial conditions and design parameters are used in the simulation: $x_1(0) = [0.5, 0]^T$, $x_2(0) = [-0.2, 0]^T$, $x_3(0) = [0.3, 0]^T$, $x_0(0) = 0$, $\hat{a}_i(0) = 0, \hat{w}_i(0) = 0, \hat{a}_{0i}(0) = 0, \hat{\delta}_i(0) = 0, \hat{\theta}_i(0) = 0$, $\Gamma_{a_i} = \Gamma_{w_i} = \Gamma_{a_{0i}} = \Gamma_{\theta_i} = \Gamma_{\delta_i} = 5$, and $k = 30$.

Simulation results in figures 22-25 validate the effectiveness of the proposed distributed adaptive neural control. Figure 22 shows that all three agents follow the desired trajectory specified by the informed leader $x_0(t)$. The boundedness of the corresponding control inputs and $\hat{a}_i$ is shown in figure 23. The boundedness of parameter estimates $\hat{w}_i$,$\hat{a}_{0i}$, $\hat{\delta}_i$ as well as NN weights $\|\hat{\theta}_i\|$ are illustrated in figures 24 and 25.

A list of published work in this area is given below.

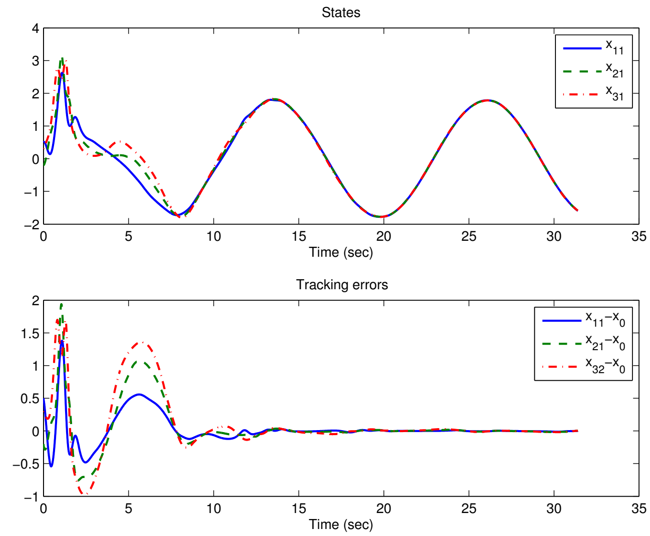

Figure 22: States and tracking errors

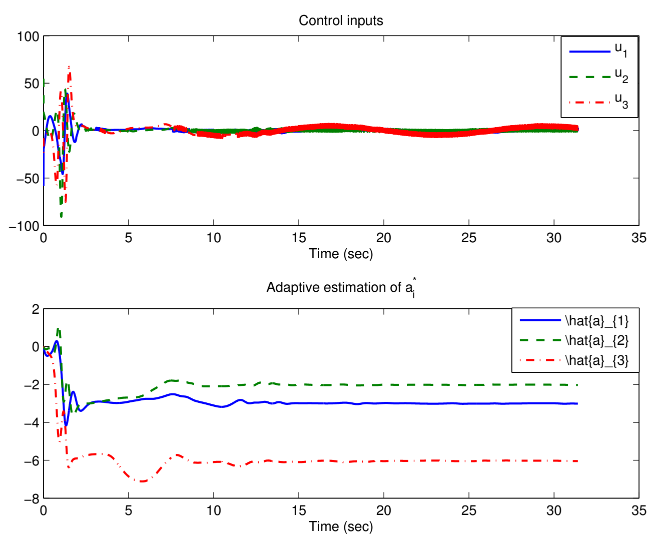

Figure 23: Control inputs and boundedness of $\hat{a}_i$

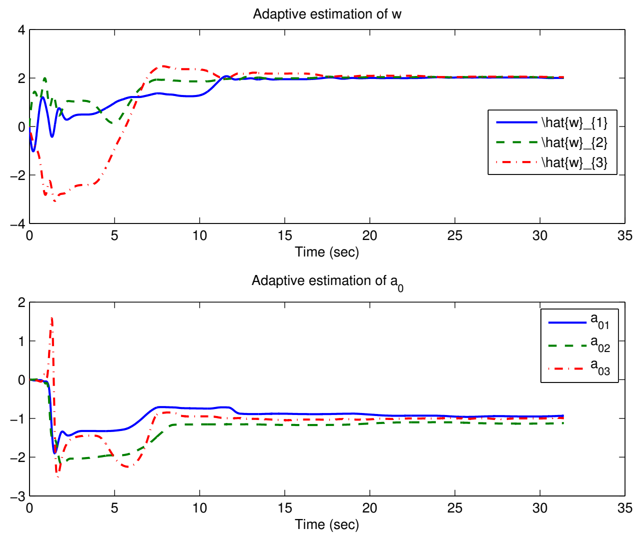

Figure 24: Boundedness of $\hat{w}_i$ and $\hat{a}_{0i}$

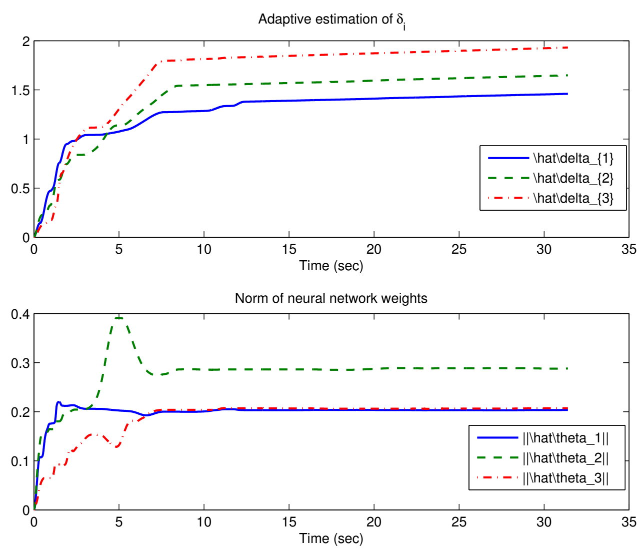

Figure 25: Boundedness of $\hat{\delta}_i$ and NN weights $||\hat{\theta}_i||$

[J-4] J. Wang, "Distributed coordinated tracking control for a class of uncertain multiagent systems, *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3423-3429, July 2017.

[C-6] J. Wang, "Adaptive cooperative control for a class of uncertain multiagent systems', *2017 American Control Conference*, Seattle, WA, May 24-26, 2017.

[C-8] J. Wang, I. S. Ahn, Y. Lu, T. Yang, and G. Staskevich, "Formation control for multiagent systems in realistic communication environments", *2017 IEEE SouthEastCon*, Charlotte NC, Mar 30-Apr. 2, 2017.

[C-9] M. Imtiaz and J. Wang, "A multiagent reinforcement learning control approach to environment exploration", *2017 IEEE SouthEastCon*, Charlotte NC, Mar 30-Apr. 2, 2017.

[C-16] J. Wang, "Distributed adaptive tracking control for a class of uncertain multiagent systems', *2016 American Control Conference*, Boston, MA, July 6-8, 2016.

5.2 Distributed gradient estimation for networked system optimization

We have also studied the distributed algorithm to address the optimization problem in multiagent systems. The aim is to find fundamental solutions to solve the problems of optimal deployment of sensor networks for distributed detection of behaviors in multiagent systems as well as dealing with complicated machine learning problems in multiagent systems.

We proposed a new distributed gradient algorithm to address the optimization problem in multiagent systems. Without loss of generality, the problem is formulated to address a general type of convex network cost function by cooperatively finding an approximately optimal solution via local computation and information exchanges among agents in the network. For practical network applications, as long as the network utility functions are properly designed via either forward or reverse engineering, the proposed distributed optimization algorithm is directly applicable to some concrete network optimization problems such as sensor network resource allocation, network flow control, TCP congestion control, and virtual network embedding. Different from most of the existing multiagent optimization results, in which the subgradient of the local cost function is the additive term in the consensus protocol, the proposed algorithm is based on the distributed estimation of the gradients of the overall network cost function. In other words, a local gradient descent algorithm is adopted by each node in the network, and the sum of gradient terms of individual cost functions is distributedly estimated via a consensus-type coordination algorithm. The proposed algorithm only requires limited communication among neighboring agents. In addition, via the estimation of the left eigenvector of a key eigenvalue of the graph Laplacian matrix, the directed communication graph can be addressed. The convergence of the proposed algorithm is rigorously analyzed under the assumption that the communication graph in the network is strongly connected. For switching communication topologies, the proposed algorithm still works by introducing a simple mechanism of periodically resetting the estimate of the left eigenvector that characterizes the communication topology. Though the proposed algorithm is an approximate one due to the use of an estimate in the gradient descent algorithm, the advantage of this algorithm is that it can be easily scaled up to address the optimization problem of very-large-scale networks with minimal increase of the computational burden and communication price because of the distributed nature of the design. Using the same framework, the constrained optimization problem is also solved by converting it into an unconstrained approximation problem via the penalty function method.

The proposed approximate distributed gradient estimation algorithm can be summarized as follows. Consider a network of n agents with the objective of finding the solution of the following optimization problem

$$\left\{ \begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} f_i(x) \\ \text{subject to} & x \in \Re^n \end{array} \right. \tag{46}$$

where $x_i \in \Re$ is the variable associated with agent i, $x = [x_1, \cdots, x_n]^T$, and $f_i : \Re^n \to \Re$ is a smooth convex function that is available only to agent i. When x_i is a vector, the treatment can be done in a similar manner. The proposed distributed gradient algorithm is of the form

$$x_i(k+1) = x_i(k) - \alpha \hat{Y}_{ii}(k) \tag{47}$$

in which $\hat{Y}_{ii}(k)$ is the estimate of $\sum_{l=1}^{n} \frac{\partial f_l(x(k))}{\partial x_i}$, given by

$$\begin{aligned} \hat{Y}_{iq}(k+1) \;\; = \;\; & \hat{Y}_{iq}(k) + \frac{1}{1+d_i} \sum_{j \in \mathcal{N}_i} a_{ij} \left[\hat{Y}_{jq}(k) - \hat{Y}_{iq}(k) \right] \\ & + \beta_i(k) \left[y_{iq}(k+1) - y_{iq}(k) \right] \end{aligned} \tag{48}$$

where $\beta_i(k) = \frac{1}{\hat{w}_{ii}(k)}$, with $\hat{w}_{ii}(k)$ being the estimate of w_i by agent i, which is updated as follows

$$\hat{w}_{ii}(k+1) = \hat{w}_{ii}(k) + \frac{1}{1+d_i} \sum_{j \in \mathcal{N}_i} a_{ij} \left[\hat{w}_{ji}(k) - \hat{w}_{ii}(k) \right] \tag{49}$$

where $\hat{w}_{ji}(k)$ is the estimate of w_i by agent j and the values are initialized as $\hat{w}_{ii}(0) = 1$ and $\hat{w}_{ji}(0) = 0, \forall j \neq i$.

The details of the proposed distributed optimization algorithm can be found in [J-1], in which the convergence proof was given. With little modification, the proposed distributed gradient estimation-based optimization algorithm can be used to solve the following optimization problem over networks

$$\left\{ \begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} f_i(x) \\ \text{subject to} & x \in \Re^m \end{array} \right. \tag{50}$$

in which n agents wish to determine an optimal consensus value $x^* \in \Re^m$, while each agent i only knows its own cost $f_i(x)$. This setup is slightly different from the problem in (46), where each agent has a state variable x_i. Nonetheless, the distributed solution can be developed following a similar procedure. If all $f_i(x)$ were available to a central node, the following centralized gradient descent algorithm could be used.

$$x(k+1) = x(k) - \alpha_k \left. \frac{\partial \sum_{i=1}^{n} f_i(x)}{\partial x} \right|_{x=x(k)} \tag{51}$$

We design a distributed algorithm based on the estimation of $\frac{\partial \sum_{i=1}^{n} f_i(x)}{\partial x}$. The following theorem states the result. Detailed proof can be found in [J-1].

Theorem 2 *Consider the distributed optimization problem in (50) over networks. Let $x^i(k) \in \Re^m$ be the estimate of optimal value x^* by agent i at the iteration step k. Define $y_i(k) = \frac{\partial f_i(x^i(k))}{\partial x}$, and let $\hat{Y}_i(k)$ be the estimate of $\sum_{i=1}^{n} y_i(k)$. Let the update law of $\hat{Y}_i(k)$ be*

$$\begin{aligned}\hat{Y}_i(k+1) &= \hat{Y}_i(k) + \frac{1}{1+d_i} \sum_{j \in \mathcal{N}_i} \left[\hat{Y}_j(k) - \hat{Y}_i(k)\right] \\ &\quad + \beta_i [y_i(k+1) - y_i(k)] \end{aligned} \tag{52}$$

where $\beta_i = \frac{1}{\hat{w}_{ii}(k)}$, with $\hat{w}_{ii}(k)$ given by (49). Then, problem (50) can be approximately solved using the following distributed optimization algorithm

$$x^i(k+1) = x^i(k) - \alpha_k \hat{Y}_i(k) \tag{53}$$

where α_k is an appropriately chosen step length.

[J-1] J. Wang and K. Pham, "An Approximate Distributed Gradient Estimation Method for Network Optimization with Limited Communications", *IEEE Trans. on SMC: Systems*, to appear, Digital Object Identifier 10.1109/TSMC.2018.2867154, Aug., 2018.

The following example illustrate the proposed algorithm in solving the resource allocation problem in multiagent systems.

Example:

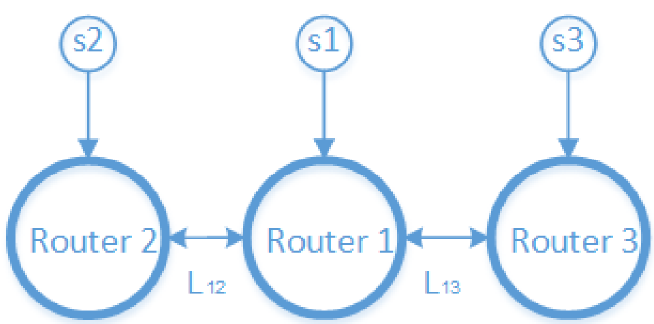

Figure 26: A simple network with three sources

As shown in figure 26, the example considers a network with three sources s_1, s_2 and s_3 competing for resources, There are three flows with the corresponding date rates x_1, x_2 and x_3. Link L_{12} between router 1 and router 2 has a capacity of 2 units per second, and link L_{13} between router 1 and router 3 has a capacity of 1 unit per second. The resource allocation problem is formulated as finding the optimal data rates such that the following utility function is maximized while satisfying the link constraints. That is,

$$\begin{cases} \text{max} & \sum_{i=1}^{3} \log(x_i) \\ \text{subject to} & x_1 + x_2 \leq 2, \quad x_1 + x_3 \leq 1, \quad x \geq 0 \end{cases} \tag{54}$$

The problem can be solved by defining the Lagrangian

$$L(x, \lambda) = \log x_1 + \log x_2 + \log x_3 - \lambda_1(x_1 + x_2) - \lambda_2(x_1 + x_3)$$

where λ_1, λ_2 are Lagrange multipliers. Based on the necessary condition for optimality and setting $\frac{\partial L}{\partial x_i} = 0$, the optimal solution can be found as follows

$$x_1^* = 0.422, \quad x_2^* = 1.577, \quad x_3^* = 0.577$$

In the following, we will show how the proposed distributed optimization algorithm can be used to solve the problem in (54). We assume that the three sources exchange information according to the connectivity topology defined in figure 27. The adjacency matrix, degree matrix and Laplacian matrix are

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \; D = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and

$$L = D - A = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

respectively. The corresponding F matrix is

$$F = I - (I + D)^{-1}L = \begin{bmatrix} 0.3333 & 0.3333 & 0.3333 \\ 0.5000 & 0.5000 & 0 \\ 0.5000 & 0 & 0.5000 \end{bmatrix}$$

and its left eigenvector, corresponding to the eigenvalue of 1, is

$$w = \begin{bmatrix} 0.4286 & 0.2857 & 0.2857 \end{bmatrix}^T$$

Although w is unknown to all sources, its estimate can be obtained using (49).

Using the penalty function method, the maximization problem in (54) can be recast as the unconstrained approximation problem defined below

$$\text{minimize} \; \sum_{i=1}^{n} f_i(x) \tag{55}$$

where $f_1(x) \triangleq -\log(x_1)$, $f_2(x) \triangleq -\log(x_2) + \frac{\gamma}{(2-x_1-x_2)}$, and $f_3(x) \triangleq -\log(x_3) + \frac{\gamma}{(1-x_1-x_3)}$, where $\gamma > 0$ is a small constant. In the simulation, $\alpha_k = 10^{-3}/\sqrt{k}$, $\gamma = 10^{-4}$, $x_1(1) =$

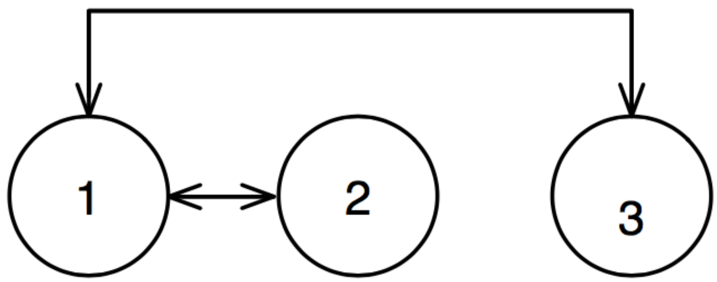

Figure 27: Information exchange topology

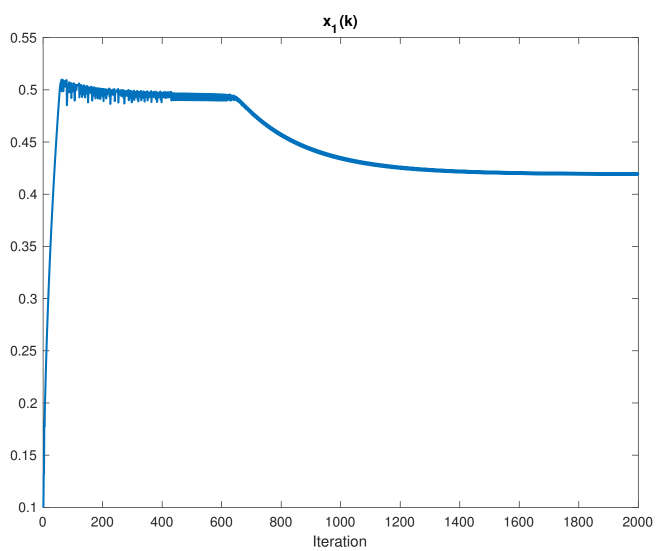

Figure 28: Convergence of x_1

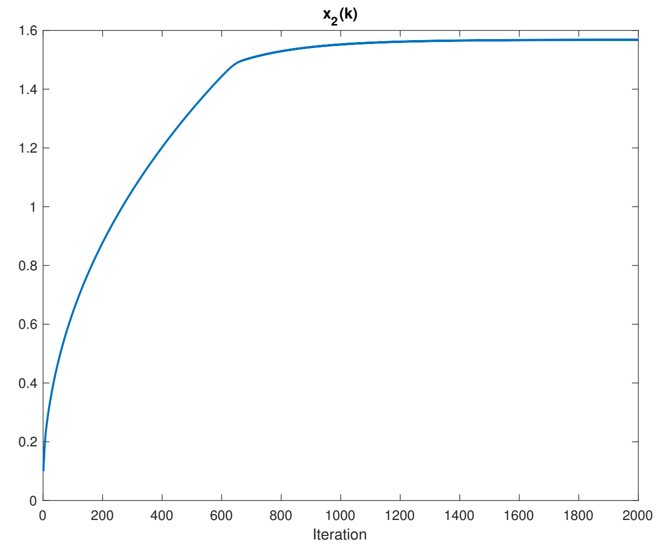

Figure 29: Convergence of x_2

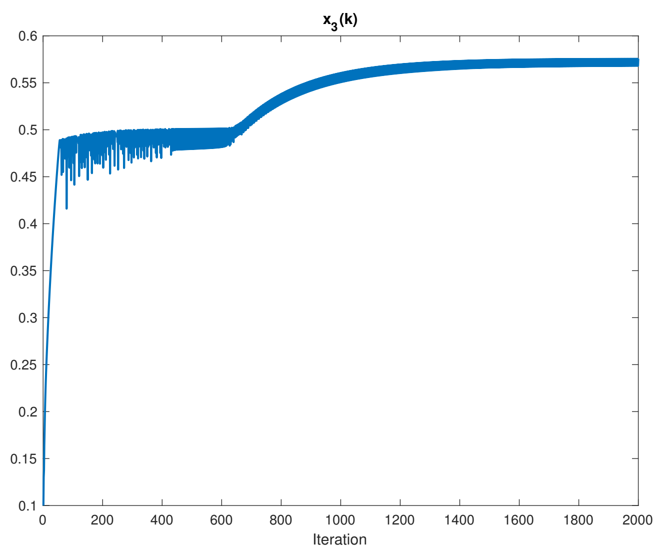

Figure 30: Convergence of x_3

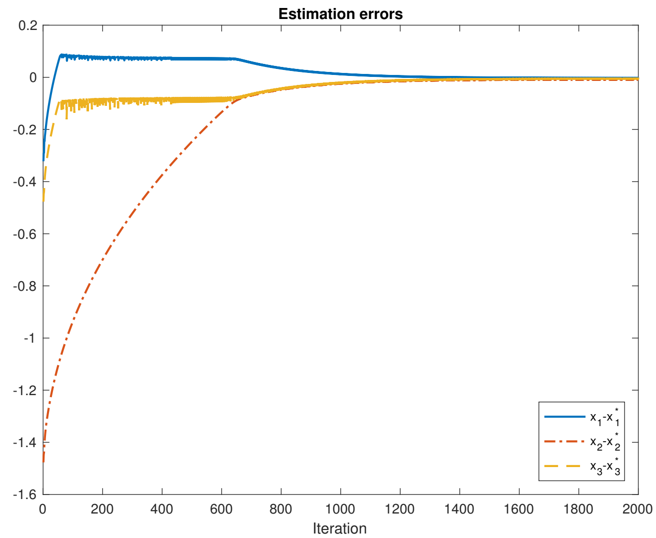

Figure 31: Estimation errors

$0.1, x_2(1) = 0.1, x_3(1) = 0.1$, and $\hat{Y}_{iq}(1) = 0$, $i = 1, 2, 3, q = 1, 2, 3$. Figures 28, 29, 30 depict the updates of $x_1(k)$, $x_2(k)$, and $x_3(k)$, respectively. Figure 31 shows the corresponding estimation errors. It can be seen that those values converge to the true optimal data rates.

The convergence rate and estimation error are dependent on the step length α_k as well as the parameter γ in the use the penalty function method. In general, a too small or a too large step length may render slow convergence and large estimation error. Specifically, if bounds of the eigenvalues of the Hessian matrix $\nabla^2 \sum_i f_i(x(k))$ is known as L_k, then the step length can be selected as $\alpha_k \leq \frac{1}{L_k}$. In addition, α_k needs to be diminishing in order to ensure the convergence of (47). However, if α_k is diminishing too fast, the convergence speed may become slow. Figure 32 shows the objective function value resulting from the first 2000 iterations under different step sizes given $\gamma = 10^{-4}$. It can be seen that the medium step length $\alpha_2 = 10^{-3}$ produces the best result in terms of convergence rate and estimation error.

Simulation was also conducted for the case of different γ. Generally, for small γ the solutions to the constrained optimization problem in (54) and the unstrained optimization problem in (55) will be nearly equal. However, as illustrated in figure 33 for the objective values resulting from the first 2000 iterations under the same $\alpha_k = 10^{-3}$ and different γ, the medium $\gamma_2 = 10^{-4}$ produces the best result. Shown in Table 1 are the results of employing several sets of different α_k and γ, where RMSE(k) is the root mean square estimation error value (RMSE) for all agents at step k defined as

$$\text{RMSE}(k) = \sqrt{\sum_{i=1}^{3} \frac{(x_i(k) - x_i^*)^2}{3}}$$

Interestingly, we can see that the combination of $\alpha_k = 2 \times 10^{-3}$ and $\gamma = 10^{-4}$ produces the best result.

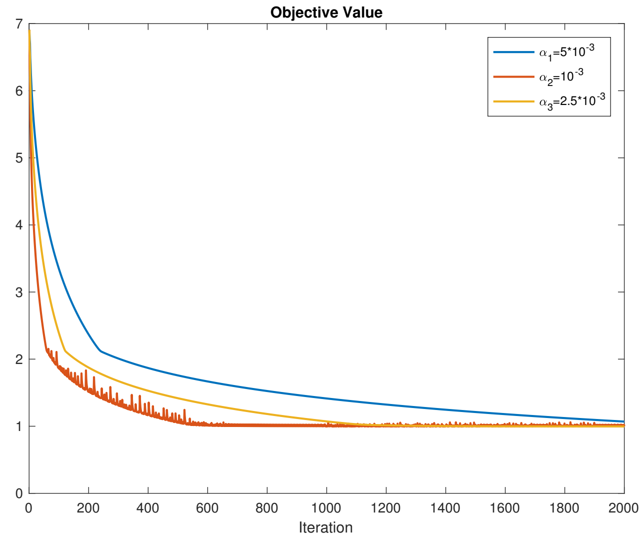

Figure 32: Objective values resulting from different step sizes

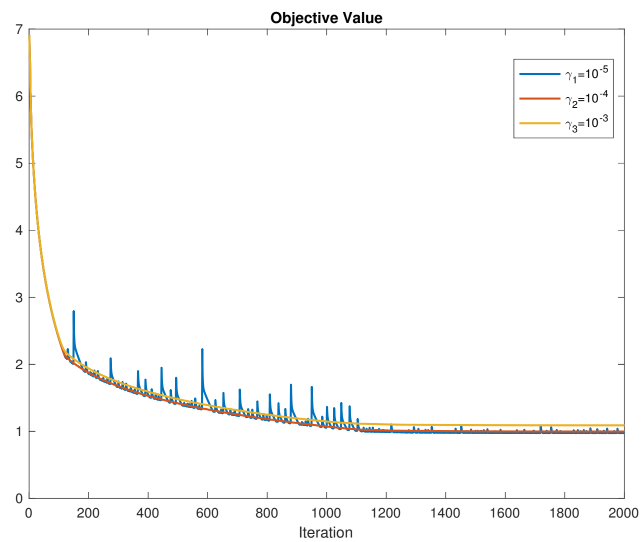

Figure 33: Objective values resulting from different γ

Table 1: RMSE value at step $k = 2000$

RMSE	$\alpha_k = 5 \times 10^{-4}$	$\alpha_k = 10^{-3}$	$\alpha_k = 2 \times 10^{-3}$
$\gamma = 10^{-4}$	0.1852	0.0164	0.0106
$\gamma = 10^{-3}$	0.1901	0.0367	0.0317
$\gamma = 10^{-5}$	0.2802	0.7616	0.9935

6.0 RESULTS AND DISCUSSION: SIMULATION AND EXPERIMENTS

The developed distributed estimation, control and optimization algorithms out of this research project have been validated through extensive software simulations and experimental tests using mobile robots and quadrotor UAVs. We first applied AgentFly software platform developed by Czech Technical University (CTU) to illustrate the UAV emergent behaviors. Matlab and Java simulation results were also developed for formation navigation of multiagents with unicycle dynamics. Experimental tests were further conducted using a number of different robot platforms including UGVs (Kilobots and Q-Bot2s at Bradley Univ.), UUVs (Eco-Dolphin at Embry-Riddle Univ.), and UAVs (Crazyfiles at Bradley Univ.).

Particularly, the distributed algorithm was implemented using several nano quadcopters called Crazyflies (Fig. 34) to control their flying behaviors. The control system contains two modules. The high-level control module is applied to generate the desired way-points for individual Crazyflies, and the low-level control module is used for stabilizing control of Crazyflies to follow trajectories formed by the way-points. The overall control algorithms were implemented with the aid of a python-based library developed by Bitcraze. Specifically, localization of Crazyflies was handled by an anchor-tag system known as the Loco Positioning System (LPS). A remote computer was used to run the high-level control laws and to transmit control signals to Crazyflies through a Crazyradio PA dongle. The radio has the ability to send multiple radio control signals to several Crazyflies at once, which allows us to update each Crazyflie's position individually. The on-board microcontroller of the Crazyflie handles the low-level PD and PID control algorithms for quick response and stabilization. Experiments including hovering control, trajectory following, and formation control were successfully conducted in an indoor environment.

Figure 34: A Crazyflie 2.0 Quadrotor UAV

The nano Crazyflie quadrotor is selected over the others on the market for experiments because of its high quality components. This quadrotor is open source, and we are able to program and control it using and adapting its python API. In addition, it is agile and safe to fly indoor. It has

5-10 minutes of flight time with less than an hour charge time. Bitcraze also produces its own Crazyradio which can be used to control multiple Crazyflies at once. Its specifications are given below.

- Weight: 27g
- Size (WxDxH): 92x92x29mm (motor-to-motor and including motor mount feet)
- 20 dBm radio amplifier tested to more than 1km range LOS with Crazyradio PA
- STM32F405 main application MCU (Cortex-M4, 168MHz, 192kb SRAM, 1Mb flash)
- nRF51822 radio and power management MCU (Cortex-M0, 32Mhz, 16kb SRAM, 128kb flash)
- IMU: 3-axis gyro, accelerometer, and magnetometer
- Max recommended payload weight: 15g

While the on-board MCU is handling the local low-level feedback control of the Crazyflie, the high-level control strategy can be programmed and implemented through a remote computer, and then the input command can be transmitted to Crazyflie through the Crazyradio PA module. This hardware/software architecture allows us seamlessly implement the proposed distributed hierarchical control strategy.

Quadrotor Dynamics

The dynamics of quadrotor UAV has been well documented in literature. A simple model is given as follows

$$\begin{aligned}
\ddot{x} &= (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{U_1}{m} \\
\ddot{\phi} &= \dot{\theta}\dot{\psi}(\frac{I_y - I_z}{I_x}) - \frac{J_R}{I_x}\dot{\theta}\Omega_R + \frac{L}{I_x}U_2 \\
\ddot{y} &= (\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi)\frac{U_1}{m} \\
\ddot{\theta} &= \dot{\phi}\dot{\psi}(\frac{I_z - I_x}{I_y}) - \frac{J_R}{I_y}\dot{\phi}\Omega_R + \frac{L}{I_y}U_3 \\
\ddot{z} &= -g + (\cos\phi\cos\theta)\frac{U_1}{m} \\
\ddot{\psi} &= \dot{\phi}\dot{\theta}(\frac{I_x - I_y}{I_z}) + \frac{1}{I_z}U_4
\end{aligned} \tag{56}$$

where x, y, z are the position of the center of the mass in the inertial frame; ϕ, θ, ψ are the Euler angles, which describe the orientation of the body-fixed frame with respect to the inertial frame;

m, I_x, I_y, and I_z are the mass and moments of inertia of the quadrotor, respectively; L is the length from the rotors to the center of mass; and J_R and Q_R are the moments of inertia and angular velocity of the propeller blades. U_1, U_2, U_3, and U_4 are the collective, roll, pitch, and yaw forces generated by the four propellers.

The desired position and orientation (pose) of a quadrotor is hovering. Accordingly, a linearized model can be obtained through Jacobian linearzation using small angle approximations. Let $U_1 = mg + \Delta U_1$. Under these approximations, equation (56) becomes

$$\begin{aligned} \ddot{x} = g\theta,\ \ddot{\phi} = \frac{L}{I_x}U_2,\ \ddot{y} = -g\phi \\ \ddot{\theta} = \frac{L}{I_y}U_3,\ \ddot{z} = \frac{\Delta U_1}{m},\ \ddot{\psi} = \frac{1}{I_z}U_4 \end{aligned} \tag{57}$$

The low-level control can be done based on the linearized model in (57).

Distributed Hierarchical Control Design

To facilitate the control design, we propose a distributed hierarchical control strategy. In other words, the desired flight patterns (trajectories) for individual UAVs are generated by a high-level controller, and a low-level control is used to generate roll, pitch and yaw input commands. The low-level control is based on linearized model in (57), which is a typical double integrator model, and its control can be done using a linear controller such as a PID controller. A simulink model for PID hovering control is given in Figure 35.

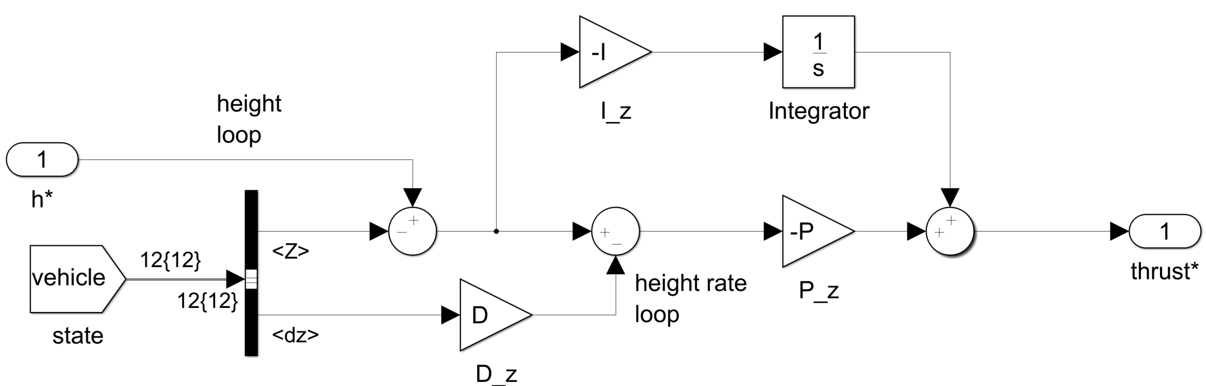

Figure 35: Hovering Control Using PID

In what follows, we focus on the design of a high-level controller. For a single UAV, it is straightforward to specify way-points and/or flight trajectories in the high-level controller. For formation flying of multiple UAVs, the desired trajectories and flight patterns will be generated through information sharing among UAVs.

Let $x_i(k), y_i(k), z_i(k)$ be the current position of UAV i at the time instant k, which is obtained through a localization system (the LOCO position system to be described in section III). Let $x_i^d(k), y_i^d(k), z_i^d(k)$ denote the desired position of UAV i at the time instant k. The high-level distributed control algorithms for generating $x_i^d(k), y_i^d(k), z_i^d(k)$ are given below

$$x_i^d(k+1) = \sum_{j=1}^{N} c_{ij}(k)(x_j(k) - d_j^x - x_i(k) + d_i^x) \tag{58}$$

$$y_i^d(k+1) = \sum_{j=1}^{N} c_{ij}(k)(y_j(k) - d_j^y - y_i(k) + d_i^y) \tag{59}$$

$$z_i^d(k+1) = \sum_{j=1}^{N} c_{ij}(k)(z_j(k) - z_i(k)) \tag{60}$$

where (d_i^x, d_i^y) are some given constants which define the formation shape, and $c_{ij}(t) > 0$ denotes that UAV j can transmit data to UAV i, otherwise $c_{ij}(t) = 0$.

LOCO Positioning System

The LOCO Positioning System is used for acquiring $x_i(k), y_i(k)$ and $z_i(k)$. It was developed by Bitcraze to act as a sort of indoor GPS system for the Crazyflie. There are six anchors that are placed throughout the space as shown in Figure 36. In addition to the anchors is a LOCO Deck, which sits on top of the Crazyflie. This deck communicates with the anchors via 2.4 GHz radio waves. From this communication, the Crazyflie is able to estimate its position in a 3D space. Bitcraze provides a Python API Library which can be applied in algorithms development of controlling Crazyflies. This library is developed to take care of the low-level control of the Crazyflie, and allowing for easier implementation of custom control algorithms.

Square Pattern Flight

Desired set-points were input to the sequence. Each set-point was sent every 0.1 seconds for 5 seconds, then the next set-point was sent. The set-point needed to be sent every 0.1 seconds so that the watchdog timer did not disable the motors.

For each test run, the actual x, y, and z values were output to the screen and to a csv file twice a second. The csv file was imported into MATLAB for analysis. These values were used to compare the desired path to the actual path the Crazyflie took. We conducted around 50 runs for a Monte-Carlo analysis. The 3D plot of all the testing flights are overlaid as shown in Figure 37. A quite robustness performance can be observed.

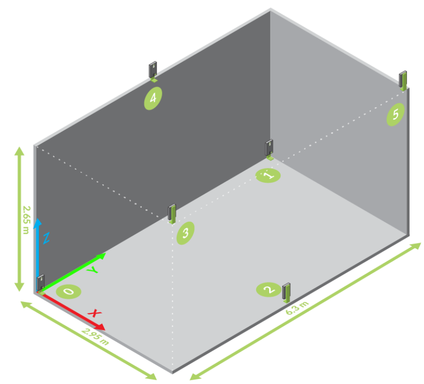

Figure 36: LOCO Positioning System Setup

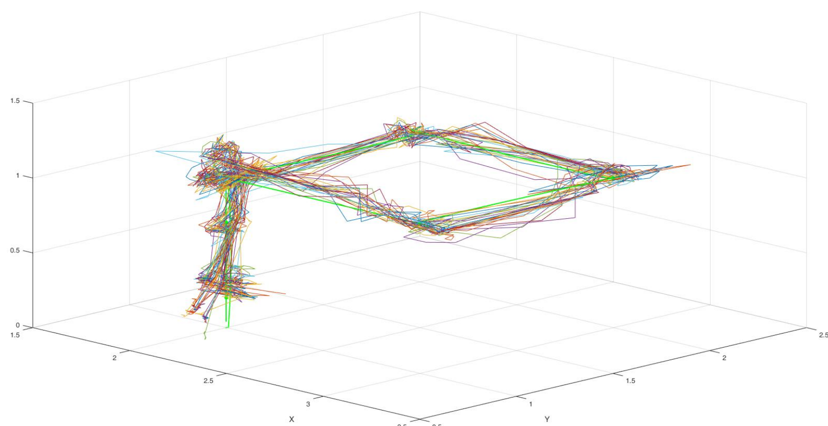

Figure 37: MATLAB Plot of the Square Trajectory Data

Circle Pattern Flight

The circle pattern was also tested The desired circular trajectory was generated and new coordinates were sent to the Crazyflie every 0.25 seconds. The circular flight was certainly more stable than the square flight because consecutive set-points were much closer together. During the square flight, set-points were a meter apart, which resulted in significant overshoot during each transition of the sequence. With the new set-points being roughly 3 centimeters apart, the Crazyflie moves much more stably and smoothly.

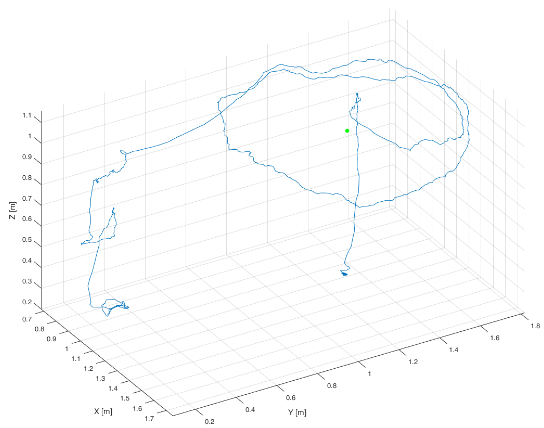

Figure 38: MATLAB Plot of the Circle Trajectory Data

After every flight, square and circular, measurements were taken to determine the landing error. The measurements were the distances between the desired landing point and the actual landing point in both the x and y directions.

Formation Flying

We have also conducted simulation for formation flying. Five crazyflies are coordinated to move into a pyramid formation while circling around the same center. Videos for experiment results can be found at our project web-page at `http://ee.bradley.edu/projects/proj2018/crazy/videos/videos.html`.

The relevant publications are listed below.

[J-5] G. Bock, R. Hendrickson, J. Lamkin, B. Dhall, J. Wang, and I. S. Ahn, “Experiments of distributed control for multiple mobile robots with limited sensing/communication capacity”, *International Journal of Handheld Computing Research*, vol. 8, no. 2, pp.19-40, April-June 2017.

[C-2] A. Le, R. Clue, J. Wang, and I. S. Ahn, “Distributed vision-based target tracking control using multiple mobile robots”, *18th IEEE International Conference on Electro Information Technology (EIT)*, Rochester, Michigan, May 3-5, 2018.

[C-3] Bryce Mack, Chris Noe, Trevor Rice, J. Wang, and I. S. Ahn, “Distributed Hierarchical Control of Quadrotor UAVs: designa and experimental validation”, *2018 IEEE SouthEastCon*, Tampa, FL, April 19 - April 22, 2018.

[C-5] H. Liu, R. Cheng, T. Yang, and J. Wang, "Modeling and verifying the communication and control of a fleet of collaborative autonomous underwater vehicles," *43th Annual Conf. of the IEEE Industrial Electronics Society (IECON17)*, Beijing, Oct. 29-Nov.1, 2017.

[C-9] M. Imtiaz and J. Wang, "A multiagent reinforcement learning control approach to environment exploration", *2017 IEEE SouthEastCon*, Charlotte NC, Mar 30-Apr. 2, 2017.

[C-14] T. A. Khan and J. Wang, "On formalization of emergent behaviors in multiagent systems with limited interactions", *2016 IEEE International Conference on Electro Information Technology (EIT)*, Grand Forks, ND, May 19-21, 2016.

[C-15] G. Bock, R. Hendrickson, J. Lamkin, B. Dhall, J. Wang and I. S. Ahn, "Experiments of distributed control for multiple mobile robots with limited sensing/communication capacity", *2016 IEEE International Conference on Electro Information Technology (EIT)*, Grand Forks, ND, May 19-21, 2016.

[C-17] H. Liu, T. Yang, and J. Wang, "Model checking for the fault tolerance of collaborative AUVs", *17th IEEE Int. Symposium on High Assurance Systems Engineering*, Orlando, FL, Jan 7-9, 2016.

7.0 CONCLUSIONS

In this report, we summarized the major research outcomes obtained through this grant. By seeking the deep understanding of interaction mechanism in multiagent systems, we developed a number of sound results in terms of distributed detection and control of emergent behaviors in multiagent systems. Particularly, we presented key components of the proposed distributed estimation, control, and optimization algorithms for multiagent systems as well as examples and simulation results to illustrate the effectiveness of the proposed designs. More details, convergence analysis, proofs, simulations, and experimental results related to the developed results are well documented in our published papers, which are submitted together with this report for ease of references.

8.0 REFERENCES

[1] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[2] D. V. Dimarogonal and K. J. Kyriakopoulos. On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on Automatic Control*, 52:916–922, 2007.

[3] R. A. Freeman, P. Yang, and K.M.Lynch. Stability and convergence properties of dynamic consensus estimators. In *Proc. IEEE Int. Conf. Decis. Control*, pages 383–343, 2006.

[4] S. Kar and J. M. F. Moura. Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise. *IEEE Trans. on Signal Processing*, 57:355–369, 2009.

[5] U. A. Khan and J. M. F. Moura. Distributed the kalman filter for large-scale systems. *IEEE Trans. on Signal Processing*, 56:4919–4935, 2008.

[6] N. E. Leonard, T. Shen, and et.al. T. Nabet. Decision versus compromise for animal groups in motion. *Proceedings of the National Academy of Sciences*, 109:227–232, 2012.

[7] M. E. J. Newman. *Network An Introduction*. Oxford Univ. Press, Oxford, 2010.

[8] R. Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Proc. IEEE Int. Conf. Decis. Control*, pages 8179–8184, 2005.

[9] R. Olfati-Saber. Distributed kalman filtering for sensor networks. In *46th IEEE Conf. on Decision and Control*, New Orleans, LA, Dec 12-14 2007.

[10] Z. Qu. *Cooperative Control of Dynamical Systems*. Springer-Verlag, London, 2009.

[11] W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer-Verlag, London, 2008.

[12] R. O. Saber, J. Alex Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95:215–233, 2007.

[13] G. Scutari and S. Barbarossa. Distributed consensus over wireless sensor networks affected by multipath fading. *IEEE Trans. on Signal Processing*, 56:4100–4106, 2008.

[14] G. Scutari, S. Barbarossa, and L. Pescosolido. Distributed decision through self-synchronizing sensor networks in the presence of propagation delays and asymmetric channels. *IEEE Trans. on Signal Processing*, 56:1667–1684, 2008.

[15] D. D. Siljak. *Large-scale dynamic systems: stability and structure*. North-Holland, New York, 1978.

[16] D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Distributed sensor fusion using dynamic consensus. In *IFAC World Congress*, Prague, Czech, 2005.

[17] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, Boston, 2004.

[18] S. H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[19] S. H. Strogatz. *SYNC: The Emerging Science of Spontaneous Order*. Hyperion, New York, 2003.

[20] T. Vicsek, A. Czirok, E. B. Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75:1226–1229, 1995.

[21] J. Wang, Z. Qu, and M. Obeng. A distributed cooperative steering control with application to nonholonomic robots. In *49th IEEE Conf. on Dec. and Ctrl*, pages 4571–4576, Atlanta, GA, Dec 2010.

[22] G. Weiss. *Multiagent Systems a Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts, 1999.

[23] A. T. Winfree. *The geometry of biological time*. Springer-Verlag, New York, 1980.

APPENDIX A – List of Publications

This multiple-year project leads to twenty-three research publications in the internationally refereed journals and conferences, which are listed below.

Journal Papers

[J-1] J. Wang and K. Pham, "An Approximate Distributed Gradient Estimation Method for Network Optimization with Limited Communications", *IEEE Trans. on SMC: Systems*, to appear, Digital Object Identifier 10.1109/TSMC.2018.2867154, Aug., 2018.

[J-2] J. Wang, I. S. Ahn, Y. Lu, T. Yang, and G. Staskevich, "A distributed least-squares algorithm in wireless sensor networks with limited and unknown communications", *International Journal of Handheld Computing Research*, vol.8, no.3, pp. 15-36, 2017. (DOI: 10.4018/IJHCR.2017070102)

[J-3] J. Wang, I.S. Ahn, Y. Lu, T. Yang, and G. Staskevich, "A Distributed Estimation Algorithm for Collective Behaviors in Multiagent Systems with Applications to Unicycle Agents", *International Journal of Control, Automation and Systems*, Springer, vol.15, no.6, pp. 2829-2839, 2017.

[J-4] J. Wang, "Distributed coordinated tracking control for a class of uncertain multiagent systems, *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3423-3429, July 2017.

[J-5] G. Bock, R. Hendrickson, J. Lamkin, B. Dhall, J. Wang, and I. S. Ahn, "Experiments of distributed control for multiple mobile robots with limited sensing/communication capacity", *International Journal of Handheld Computing Research*, vol. 8, no. 2, pp.19-40, April-June 2017.

[J-6] J. Wang, T. Yang, G. Staskevich, and B. Abbe, "Approximately Adaptive Neural Cooperative Control for Nonlinear Multiagent Systems with Performance Guarantee", *International Journal of Systems Science*, Taylor & Francis, vol. 48, no. 5, pp. 909-920, 2016.

Conference Papers

[C-1] J. Wang, I. S. Ahn, Y. Lu, T. Yang, and G. Staskevich, "Resilient detection of multiple targets using a distributed algorithm with limited information sharing", *2018 SPIE Defense Conference*, Orlando, FL, April 16 – April 19, 2018.

[C-2] A. Le, R. Clue, J. Wang, and I. S. Ahn, "Distributed vision-based target tracking control using multiple mobile robots", *18th IEEE International Conference on Electro Information Technology (EIT)*, Rochester, Michigan, May 3-5, 2018.

[C-3] Bryce Mack, Chris Noe, Trevor Rice, J. Wang, and I. S. Ahn, "Distributed Hierarchical Control of Quadrotor UAVs: designa and experimental validation", *2018 IEEE SouthEastCon*, Tampa, FL, April 19 - April 22, 2018.

[C-4] J. Wang, "Distributed estimation of moving targets with unknown dynamics ", *2018 IEEE Aerospace Conference*, Big Sky, MT, March 3-10, 2018.

[C-5] H. Liu, R. Cheng, T. Yang, and J. Wang, "Modeling and verifying the communication and control of a fleet of collaborative autonomous underwater vehicles," *43th Annual Conf. of the IEEE Industrial Electronics Society (IECON17)*, Beijing, Oct. 29-Nov.1, 2017.

[C-6] J. Wang, "Adaptive cooperative control for a class of uncertain multiagent systems', *2017 American Control Conference*, Seattle, WA, May 24-26, 2017.

[C-7] J. Wang, I. S. Ahn, Y. Lu, T. Yang, and G. Staskevich, "A distributed least-squares algorithm in wireless sensor networks with limited communication", *17th IEEE International Conference on Electro Information Technology (EIT)*, Lincoln, Nebraska, May 14-17, 2017.

[C-8] J. Wang, I. S. Ahn, Y. Lu, T. Yang, and G. Staskevich, "Formation control for multiagent systems in realistic communication environments", *2017 IEEE SouthEastCon*, Charlotte NC, Mar 30-Apr. 2, 2017.

[C-9] M. Imtiaz and J. Wang, "A multiagent reinforcement learning control approach to environment exploration", *2017 IEEE SouthEastCon*, Charlotte NC, Mar 30-Apr. 2, 2017.

[C-10] J. Wang, I. S. Ahn, Y. Lu, and G. Staskevich, A New Distributed Algorithm for Environmental Monitoring by Wireless Sensor Networks with Limited Communication, *2016 IEEE Sensors*, Orlando, FL, Oct 30 Nov 2, 2016.

[C-11] Z. Fu, T. Yang, and J. Wang, Effects of Jamming on a Multi-agent Flocking Model with Distributed Estimation of Global Features, *SAI Intelligent Systems Conference 2016*, San Francisco, CA, Dec 6-7, 2016.

[C-12] J. Wang, I. S. Ahn, Y. Lu, and T. Yang, A distributed detection algorithm for collective behaviors in multiagent systems, *the 12th World Congress on Intelligent Control and Automation*, Guilin, China, June 12-15, 2016.

[C-13] T. Yang, Z. Fu, and J. Wang, Application of an even-triggered distributed estimation algorithm in a simple multiagent flocking model, *2016 IEEE SoutheastCon*, Norfolk VA, March 30-Apr 3, 2016.

[C-14] T. A. Khan and J. Wang, "On formalization of emergent behaviors in multiagent systems with limited interactions", *2016 IEEE International Conference on Electro Information Technology (EIT)*, Grand Forks, ND, May 19-21, 2016.

[C-15] G. Bock, R. Hendrickson, J. Lamkin, B. Dhall, J. Wang and I. S. Ahn, "Experiments of distributed control for multiple mobile robots with limited sensing/communication capacity", *2016 IEEE International Conference on Electro Information Technology (EIT)*, Grand Forks, ND, May 19-21, 2016.

[C-16] J. Wang, "Distributed adaptive tracking control for a class of uncertain multiagent systems', *2016 American Control Conference*, Boston, MA, July 6-8, 2016.

[C-17] H. Liu, T. Yang, and J. Wang, "Model checking for the fault tolerance of collaborative AUVs", *17th IEEE Int. Symposium on High Assurance Systems Engineering*, Orlando, FL, Jan 7-9, 2016.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

UAV	Unmanned Aerial Vehicle
UGA	Unmanned Ground Vehcle
RMSE	Root Mean Square Estimation Error
WSN	Wireless Sensor Network
PID	Proportional-Integral-Derivative
RBF NN	Radial Basis Function Neural Network
$\triangleq$	defined as
$<$ $(>)$	less (greater) than
$\leq$ $(\geq)$	less (greater) than or equal to
$\forall$	for all
$\in$	belongs to
$\rightarrow$	tends to
$\sum$	summation
$\bigcup$	union
$\|x\|$	the norm of a vector x
$\max$	maximum
$\min$	minimum
$\Re^n$	the $n-$dimensional Euclidean space
diag$[x_1, \cdots, x_n]$	a diagonal matrix with diagonal elements x_1 to x_n
$\dot{x}$	the first derivative of x with respect to time
$\ddot{x}$	the second derivative of x with respect to time
$A^T(x^T)$	the transpose of a matrix A (a vector x)
argmin	the argument of the minimum
$\mathcal{L}_2$	the space defined based on $2-$norm
$\mathcal{L}_\infty$	the space defined based on $\infty-$norm
sgn$(\cdot)$	the signum function