Chinese Journal of Aeronautics, (2015),28(1): 191–199

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

Consensus based on learning game theory with a UAV rendezvous application

Lin Zhongjie, Liu Hugh hong-tao *

University of Toronto Institute for Aerospace Studies, 4925 Dufferin Street, Toronto, Ontario M3H 5T6, Canada

Received 23 July 2014; revised 15 September 2014; accepted 2 November 2014
Available online 26 December 2014

KEYWORDS

Consensus;
Distributed algorithms;
Fictitious play;
Game theory;
Multi-agent systems;
Potential game

Abstract Multi-agent cooperation problems are becoming more and more attractive in both civilian and military applications. In multi-agent cooperation problems, different network topologies will decide different manners of cooperation between agents. A centralized system will directly control the operation of each agent with information flow from a single centre, while in a distributed system, agents operate separately under certain communication protocols. In this paper, a systematic distributed optimization approach will be established based on a learning game algorithm. The convergence of the algorithm will be proven under the game theory framework. Two typical consensus problems will be analyzed with the proposed algorithm. The contributions of this work are threefold. First, the designed algorithm inherits the properties in learning game theory for problem simplification and proof of convergence. Second, the behaviour of learning endows the algorithm with robustness and autonomy. Third, with the proposed algorithm, the consensus problems will be analyzed from a novel perspective.

1. Introduction

Multi-agent cooperation problems are becoming more and more attractive in both civilian and military applications, such as the real-time monitoring of crops status, monitoring the traffic situation,[1,2] forest fire monitoring,[3,4] surveillance[5–7] and battle field assessment.[8] In multi-agent cooperation problems, different network topologies will influence different manners of cooperation between agents. A centralized system will directly control the operation of each agent with information flow from a single centre, while in a distributed system, agents operate separately under certain communication protocols.

There are many related studies on distributed multi-agent cooperation problems such as flocks,[9] swarms,[10,11] sensor fusion[12,13] and so on. The theoretical framework for constructing and solving consensus problems within networked systems was introduced in Refs. [14–17].

For distributed multi-agent systems, there are three challenges need to be addressed to achieve cooperation among a potentially large number of involving agents:

* Corresponding author. Tel.: +1 416 667 7928.
E-mail address: liu@utias.utoronto.ca (H.h.-t. Liu).

Peer review under responsibility of Editorial Committee of CJA.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.cja.2014.12.009
1000-9361

(1) Limited information for agents to utilize to achieve the global objective.
(2) The information of the distributed network topology is unknown to agents.
(3) The network can be dynamic, such as stochastic networks or switching networks.

To construct a game theory model for a distributed multi-agent cooperation problem: first, the agents of the game and their strategy domains needs to be identified, in which agents are considered as myopic and rational players; second, based on the strategy domains and the mission requirement, a global cost function that reflects the performance of the joint actions will be constructed; third, an optimal strategy will be derived out based on the established global cost function. Therefore, in order to achieve cooperation in a distributed network among myopic agents, the designed game will yield the lowest costs for all agents if and only if when their strategies will benefit the global objective.

Based on bargaining game concept, the author suggests a useful strategy for task assignment and resource distribution problems with a small number of agents.[18] An approach is proposed for consensus problems based on cooperative game theory and bargain game theory.[19] A distributed optimization algorithm is designed based on the state potential game framework.[20,21]

In classical game theory, agents are assumed to be able to correctly anticipate what their opponents will do. It may be unrealistic in practice. In real application, agents cannot introspectively anticipate how others will act in distributed networks, especially in games with a large number of participants. Unlike classical game theory, learning game theory does not impose assumptions on agents' rationality and believes, but assumes instead that agents can learn over time about the game and the behaviour patterns of their opponents.

The preliminary result was presented.[22] In this paper, a systematic distributed optimization approach will be established based on a learning game algorithm. The convergence of the algorithm will be proven under game theory framework. Two typical consensus problems will be analyzed with the proposed algorithm.

2. Preliminaries

For a game $\mathcal{G}(N,(\mathcal{U}_i, i\in N),(J_i, i\in N))$, $\mathcal{U}_i$ represents agent i's strategy domain with its element as μ_i. The collected strategy domain of $\mathcal{U}_i$ is denoted as $\mathcal{U}=\times_{i\in N}\mathcal{U}_i$ with its element as $\mu=(\mu_i,\mu_{-i})$, where μ_{-i} denotes the strategies of other agents besides agent i. The cost function for agent i is denoted as $J_i(\mu_i,\mu_{-i})$, which implies that agent i's cost depends both on its own strategy and on every other agent's strategy.

2.1. Governing equation

Fictitious play algorithm is adopted for designing the distributed optimization algorithm in this work. It is a variant of a "stationary Bayesian learning" process. In fictitious play, agents assume behaviour patterns of their competitors were stationary.[22]

Consider a 2×2 matrix game

$$\begin{array}{ccc} & \mu_2^1 & \mu_2^2 \\ \mu_1^1 & \left[(a_{11},b_{11}) \right. & \left. (a_{12},b_{12}) \right] \\ \mu_1^2 & \left[(a_{21},b_{21}) \right. & \left. (a_{22},b_{22}) \right] \end{array} \tag{1}$$

where $a_{ij}\ (i=1,2;\ j=1,2)$ represents the cost of Agent 1 according to its strategies $\{\mu_1^1,\mu_1^2\}$ and $b_{ij}\ (i=1,2;\ j=1,2)$ is the cost of Agent 2 for its strategies $\{\mu_2^1,\mu_2^2\}$. Agents will decide their strategies based on their current empirical frequency for their opponents. The algorithm starts with an initial empirical frequency

$$\begin{cases} e_1^2(0)=(w_1^2,w_2^2) \\ e_2^1(0)=(w_1^1,w_2^1) \end{cases} \tag{2}$$

where $w_i^j\ (i=1,2;\ j=1,2)$ represents the anticipated weight for opponent j's ith strategy.

Take Agent 1 as an example, according the empirical frequency of Agent 1, the cost for each strategy is

$$\begin{cases} J_1(\mu_1^1|e_1^2)=\dfrac{w_1^2a_{11}+w_2^2a_{12}}{w_1^2+w_2^2} \\ J_1(\mu_1^2|e_1^2)=\dfrac{w_1^2a_{21}+w_2^2a_{22}}{w_1^2+w_2^2} \end{cases} \tag{3}$$

Both agents will choose the strategies that yields the lowest cost based on their own empirical frequencies. At the end of each game, agents will update their empirical frequency according to their opponents' strategies in the previous game.

Definition 1. (ε-*dominate*[23]). A strategy $\mu_i\in\mathcal{U}_i$ is ε-dominated by another strategy $\hat{\mu}_i\in\mathcal{U}_i$, if for any $\mu_{-i}\in\mathcal{U}_{-i}$, there exists $J_i(\mu_i,\mu_{-i})>J_i(\hat{\mu}_i,\mu_{-i})+\varepsilon$. The set of ε-dominate strategy domain of agent i is denoted as

$$\mathcal{D}_i^\varepsilon(\mathcal{U})=\{\mu_i\in\mathcal{U}_i|J_i(\mu_i,\mu_{-i})+\varepsilon\leqslant J_i(\hat{\mu}_i,\mu_{-i}), \\ \forall\hat{\mu}_i\in\mathcal{U}_i,\quad \forall\mu_{-i}\in\mathcal{U}_{-i},\varepsilon\geqslant 0\} \tag{4}$$

According to the definition of the ε-dominate domain, an adaptive learning is defined as

Definition 2. (*Adaptive learning*[23]). A sequence of strategy for agent i as $\{\mu_i(t)\}$ is consistent with adaptive learning, if

$$\mu_i(t)\in\mathcal{D}_i^\varepsilon(\{\mu(\tau)|\hat{t}\leqslant\tau<t\}),\quad \forall\hat{t}<t \tag{5}$$

Definition 3. (*Nash equilibrium*). A strategy $\mu^*=(\mu_i^*,\mu_{-i}^*)$ is called a Nash Equilibrium of game $\mathcal{G}(N,\{\mathcal{U}_i,i\in N\},\{J_i,i\in N\})$, if and only if

$$J_i(\mu_i^*,\mu_{-i}^*)\leqslant J_i(\mu_i,\mu_{-i}^*),\quad \forall i\in N,\quad \forall\mu_i\in\mathcal{U}_i \tag{6}$$

Definition 4. (*Convergence*). The sequence $\{\zeta(t)\}$ converges to ζ if and only if there is a $T>0$ such that $\zeta(t)=\zeta$ for all $t\geqslant T$.

Lemma 1. (Adaptive learning convergence[23]). *If certain strategy sequence $\{\mu_i(t)\}$ is consistent with adaptive learning and it converges to μ_i^*, then μ_i^* belongs to a pure strategy Nash equilibrium $\mu^*=(\mu_i^*,\mu_{-i}^*)$.*

2.2. Potential games

The potential game concept is important for distributed multi-agent cooperation problems.

Definition 5. (*Potential game*[24]). A game $\mathcal{G}(N, \{\mathcal{U}_i, i \in N\}, \{J_i, i \in N\})$ is a potential game if there exists a global function $\Phi(\mu_i, \mu_{-i})$ such that

$$J_i(\mu_i, \mu_{-i}) - J_i(\hat{\mu}_i, \mu_{-i}) = \Phi(\mu_i, \mu_{-i}) - \Phi(\hat{\mu}_i, \mu_{-i}),$$
$$\forall i \in N, \mu_i, \hat{\mu}_i \in \mathcal{U}_i, \mu_{-i} \in \mathcal{U}_{-i} \tag{7}$$

If the strategy domain for every agent of a potential game is bounded, this game is defined as a bounded potential game.

The potential game concept provides a valuable theoretical framework for distributed multi-agent cooperation problems. First, a game admits the potential property is guaranteed to possess a Nash equilibrium. Second, from the definition of a potential game, the Nash equilibrium for every local cost function is consistent with the global objective. The potential game framework, therefore, provides distributed optimization problems with theoretical support for problem simplification. Many design methodologies[25–27] can be adopted to derive a potential game according to the global objective.

2.3. Continuous and discrete consensus algorithm

The interaction topology of a network can be denoted as $G(N, \mathcal{E})$ with the set of agents N and edges $\mathcal{E}$. The set of the neighbours of agent i is denoted as $N_i = \{j \in N | (i,j) \in \mathcal{E}\}$. An adjacent matrix $\boldsymbol{A} = [a_{ij}] \in \mathbf{R}^{n\times n}$ is defined as

$$\begin{cases} a_{ij} > 0 & \text{If } (i,j) \in \mathcal{E} \\ a_{ij} = 0 & \text{Otherwise} \end{cases}$$

The Laplacian matrix $\boldsymbol{L} = [l_{ij}] \in \mathbf{R}^{n\times n}$ is defined as that $l_{ii} = \sum_{j=1}^{n} a_{ij}$ and $l_{ij} = -a_{ij}$ if $i \neq j$.

Definition 6. (*Stochastic matrix*). A nonnegative square matrix is a row stochastic matrix if its every row is summed up to one.[28,29]

The product of two stochastic matrices is still a stochastic matrix. A row stochastic matrix $\boldsymbol{P} \in \mathbf{R}^{n\times n}$ is a stochastic indecomposable and aperiodic (SIA) matrix, if $\lim_{k\to\infty} \boldsymbol{P}^k = 1\boldsymbol{y}^{\mathrm{T}}$ with $\boldsymbol{y} \in \mathbf{R}^{n\times n}$.

The continuous-time consensus algorithm for single integrator system is described as

$$\dot{x}_i(t) = \sum_{j\in N_i} a_{ij}(t)(x_j(t) - x_i(t)) \tag{8}$$

Its Laplacian matrix form is represented as

$$\dot{\boldsymbol{x}}(t) = -\boldsymbol{L}(t)\boldsymbol{x}(t) \tag{9}$$

The consensus algorithm in Eqs. (8) and (9) is for a single-integrator system with dynamics as $\dot{x}_i = u_i$. From Eq. (9), the discrete consensus control law is derived as

$$\begin{cases} \boldsymbol{x}(k+1) - \boldsymbol{x}(k) = -\boldsymbol{L}(k)\boldsymbol{x}(k) \\ \boldsymbol{x}(k+1) = (1 - \boldsymbol{L}(k))\boldsymbol{x}(k) = \boldsymbol{A}(k)\boldsymbol{x}(k) \\ x_i(k+1) = \sum_{j\in N_i} a_{ij}(k)x_j(k) \end{cases} \tag{10}$$

The matrix $\boldsymbol{A}(k)$ is row stochastic matrix, which suggests that

$$\lim_{t\to\infty} \mathrm{e}^{-\boldsymbol{L}t} = 1\boldsymbol{v}^{\mathrm{T}} \tag{11}$$

and

$$\lim_{k\to\infty} \boldsymbol{A}^k = 1\boldsymbol{u}^{\mathrm{T}} \tag{12}$$

where $\sum_i v_i = 1$ and $\sum_i u_i = 1$. Therefore, Eqs. (11) and (12) suggest that $\lim_{t\to\infty} \boldsymbol{x}(t) = \sum_i v_i x_i(0)$ and $\lim_{t\to\infty} \boldsymbol{x}(k) = \sum_i u_i x_i(1)$.

3. Distributed optimization algorithm

Under a centralized network, the centre can obtain information from all agents and conduct the optimization. For a distributed system, only partial information is available for each agent due to the distributed network topology.

According to the mission requirement of a distributed multi-agent cooperation problem, the global cost function is defined as

$$J_{\mathrm{g}}(x_1, x_2, \ldots, x_n) \tag{13}$$

and the optimization purpose of this problem is to find optimal $\{x_i^*, i \in N\}$ that satisfies

$$x_i^* = \arg\max_{x_i \in \mathcal{U}_i} J_{\mathrm{g}}(x_1, x_2, \ldots, x_n), \quad i \in N \tag{14}$$

In a distributed system, the only information available for agent i is its own local state x_i and the information from its neighbours $\{x_j, \quad j \in N_i\}$. A distributed optimization algorithm, therefore, aims at designing a protocol under a distributed network to achieve an asymptotic convergence of the state of each agent $\{x_i(k), \quad i \in N\}$ according to the global objective in Eq. (13).

3.1. Distributed optimization algorithm based on fictitious play concept

In order to adopt and extend the fictitious play concept to establish a distributed optimization approach, first, the empirical frequency will be extended to the continuous domain. Second, based on the global objective and local empirical frequency, a local strategy evaluation function for agent i will be established. Third, an optimal strategy for agent i will then be decided for next iteration process.

3.1.1. Empirical frequency

In order to adopt the fictitious play concept for a continuous distributed problem, a distribution function is adopted to model the empirical frequency build up by each agent for its counterparts.

Consider the strategy domain $\mathcal{U}_i$ for every agent i is continuous and convex. Agent i will model the strategy adopted by its opponent j at time k as a Gaussian distribution $\mu_j(k) = f(u_j(k), \sigma_j^2(k)) \sim N(u_j(k), \sigma_j^2(k))$. The empirical frequency of agent i for its counterpart j at time k is defined as $e_i^j(k)$. If the strategy agent j plays in time $k+1$ is

$$\mu_j(k+1) = f(u_j(k+1), \sigma_j^2(k+1))$$

the empirical frequency $e_i^j(k)$ is updated as

$$e_i^j(k+1) = \frac{1}{2}(e_i^j(k) + \mu_j(k+1)) \sim N(u_{i,j}(k+1), \sigma_{i,j}^2(k+1)) \tag{15}$$

with

$$\begin{cases} u_{i,j}(k+1) = \frac{1}{2}(u_{i,j}(k) + u_j(k+1)) \\ \sigma_{i,j}^2(k+1) = \frac{1}{4}(\sigma_{i,j}^2(k) + \sigma_j^2(k+1)) \end{cases}$$

and the distribution function for $e_i^j(k)$ is denoted as $f_i(\mu_j(k))$.

Every agent $i \in N$ will establish its local empirical frequency at time k for all its opponents as $e_i(k) = \{e_i^j(k), j \in N_{-i}\}$. During each game, agent i will not only share state information but also empirical frequency with its neighbours and update with the protocol given in Eq. (15).

3.1.2. Optimization

Based on the updating and sharing protocols for the empirical frequency, the cost for strategy $\mu_i \in \mathcal{U}_i$ under empirical frequency $e_i^j(k)$ is calculated by

$$J_i^j(\mu_i | e_i^j(k)) = \int J_i^j(\mu_i, \mu_j) f_i(\mu_j(k)) \mathrm{d}\mu_j \tag{16}$$

for a two agents game problem.

For the distributed optimization problem as suggested in Eq. (13), agent i will establish its empirical frequency for all its opponents as $e_i(k) = \{f_i(\mu_1(k)), \ldots, f_i(\mu_{i-1}(k)), f_i(\mu_{i+1}(k)), \ldots, f_i(\mu_n(k))\}$. Under the empirical frequency $e_i(k)$, the cost for state x_i is derived as

$$J_i(x_i | e_i(k)) = \int \cdots \int J_g(x_1, x_2, \ldots, x_n) f_i(\mu_1(k)) \ldots f_i(\mu_{i-1}(k)) f_i(\mu_{i+1}(k)) \ldots f_i(\mu_n(k)) \, \mathrm{d}x_1 \mathrm{d}x_2 \ldots \mathrm{d}x_{i-1} \mathrm{d}x_{i+1} \ldots \mathrm{d}x_n \tag{17}$$

with $x_i \in \mathcal{U}_i$.

In every game, agent i will always search for the strategy $x_i^* \in \mathcal{U}_i$ that yields the lowest cost for Eq. (17) under current empirical frequency as

$$J_i(x_i^* | e_i(k)) \leqslant J_i(\hat{x}_i | e_i(k)), \quad \forall \hat{x}_i \in \mathcal{U}_i \tag{18}$$

and update $x_i(k+1) = x_i^*$.

The cost function $J_i(\mu_i | e_i(k))$ for strategy μ_i under the local empirical frequency $e_i(k)$ is different from the local cost function $J_i(\mu_i, \mu_{-i})$ as discussed in Eq. (7). The empirical frequency based cost function $J_i(\mu_i | e_i(k))$ is a constructed function based on the local empirical frequency $e_i(k)$ and the global cost function $J_g(x_1, x_2, \ldots, x_n)$, while the local cost function $J_i(\mu_i, \mu_{-i})$ is a predefined function based on the current strategies from agents which are accessible to agent i under a certain network topology. In this sense, without information from others, the local cost function cannot be verified, while agent i can still make decision based on its empirical frequency and $J_i(\mu_i | e_i(k))$.

3.2. Convergence analysis and remarks

Definition 2 suggests that for every iteration, if agent i's strategy is selected from the $\varepsilon -$ dominate strategy domain, this algorithm is consistent with adaptive learning procedure. For the distributed optimization algorithm introduced in the previous section, the proposed algorithm suggests that agents will always adopt strategies that yield lowest cost based on their current empirical frequency. In this sense, the proposed distributed optimization algorithm is an adaptive learning process. According to Lemma 1, the sequence of state for each agent $\{x_i(k), i \in N\}$ follows the proposed distributed optimization algorithm will converge to a pure Nash equilibrium x^* provided that the distributed problem Eq. (14) exists one.

Remark 1. From the updating protocol as defined in Eq. (15), the memory size m_i^j of $e_i^j(k)$ is defined as that agent i will only record the most recent m_i^j strategies of its opponent j as $\{\mu_j(k - m_i^j + 1), \ldots, \mu_j(k-1), \mu_j(k)\}$.

Remark 2. Since agents in learning game theory have no prior knowledge concerning their opponents, both the empirical frequency and strategies concerning their opponents are modelled as Gaussian distribution on $(-\infty, +\infty)$.

4. Application to consensus problems

In this section, the theoretical developments proposed in previous sections will be illustrated with two typical consensus problems. Based on the problem requirements, the two problems will firstly be modelled as distributed optimization problems and the proposed algorithm will be adopted for solving them. The first problem is a distributed averaging problem and the second problem is a rendezvous problem, and in which, every agent is modelled as a UAV with unicycle model. The simulation result concerning the two problems will be presented and discussed in the next section.

4.1. Distributed averaging problem

Consider there exist n agents with their initial states are not equal to each other and the sum of their states equals to $l > 0$. The state domain for each agent is continuous, convex and bounded as $\mathcal{U}_i = (0, l], \quad \forall i \in N$. The inter-agent connection is distributed, i.e. the network topology is not a full graph. The mission requirement is to design a distributed protocol to let the state of every agent reach a consensus. For this typical consensus problem, the global cost function is designed as

$$J_g(x_1, x_2, \ldots, x_n) = \sum_{i \neq j} a_{ij}(x_i - x_j)^2 \quad (i, j \in N, \ a_{ij} \geqslant 0) \tag{19}$$

where x_i and x_j represent the states of agent i and agent j, respectively. By analyzing the cost function, the only equilibrium is achieved when $x_i = x_j, \quad \forall i, j \in N$.

Based on the problem requirement, the local cost function can be designed as

$$J_i(x_i, x_{-i}) = \sum_{j \in N_i} a_{ij}(x_i - x_j)^2 \tag{20}$$

with the neighbour set for each agent is defined based on the distributed and connected network topology.

From the definition of the local cost function, the designed game is a bounded potential game[24] with its potential function as the global cost function Eq. (19). According to the properties of a potential game, the designed game is guaranteed to possess a Nash equilibrium and the equilibrium for local cost function Eq. (20) is consistent the global objective Eq. (19).

Provided that the empirical frequency $e_i^j(k)$ is defined as a Gaussian process as $f_i(\mu_j(k)) \sim N(u_{i,j}(k), \sigma_{i,j}^2(k))$, the cost for a strategy $x_i \in \mathcal{U}_i$ is

$$\begin{aligned} J_i(x_i|e_i) &= \int \dots \int \sum_{j\in N_i} a_{ij}(x_i - x_j)^2 f_i(\mu_1) f_i(\mu_2) \dots f_i(\mu_{i-1}) f_i(\mu_{i+1}) \\ &\quad \dots f_i(\mu_n) \mathrm{d}x_1 \mathrm{d}x_2 \dots \mathrm{d}x_{i-1} \mathrm{d}x_{i+1} \dots \mathrm{d}x_n \\ &= \sum_{j\in N_i} \int a_{ij}(x_i - x_j)^2 f_i(\mu_j) \mathrm{d}x_j \\ &= \sum_{j\in N_i} a_{ij}(u_{i,j}^2 + \sigma_{i,j}^2 - 2x_i u_{i,j} + x_i^2) \end{aligned} \tag{21}$$

where the time k is ignored for concision.

Because of the assumption of myopic for agents and the convexity of the local cost function, the optimal strategy x_i^* for agent i based on its empirical frequency is derived from

$$\begin{aligned} \frac{\partial J_i(x_i|e_i)}{\partial x_i} &= \frac{\partial}{\partial x_i} \sum_{j\in N_i} a_{ij}(u_{i,j}^2 + \sigma_{i,j}^2 - 2x_i u_{i,j} + x_i^2) \\ &= \sum_{j\in N_i} 2a_{ij}(x_i - u_{i,j}) = 0 \end{aligned} \tag{22}$$

Therefore,

$$x_i^* = \frac{\sum_{j\in N_i} a_{ij} u_{i,j}}{\sum_{j\in N_i} a_{ij}} \tag{23}$$

which is the weighted average of the mean values of agent i's empirical frequency.

Further, if the weight a_{ij} admits $\sum_{j\in N_i} a_{ij} = 1$ and the memory size for agent i's empirical frequency equals 1, which implies $f_i(\mu_j(k)) = f(x_j(k), \sigma_j^2(k))$, then

$$x_i(k+1) = \sum_{j\in N_i} a_{ij} x_j(k) \tag{24}$$

which is consistent with the discrete consensus protocol.[3]

Lemma 2. *Every bounded potential game has approximate finite improvement property (AFIP).*[24]

Theorem 1. *The algorithm proposed in this section will converge with finite iteration steps for the consensus problem as suggested in* Eq. (19) *and will only converge to the consensus state as*

$$\zeta^* = \{x_i^*,\ i \in N\} : x_1^* = x_2^* = x_3^* = \dots = x_n^* \tag{25}$$

Proof. The proof of this theorem is based on the proof of the following two statements:

(1) The sequence of this algorithm will converge with finite iteration steps
(2) If the algorithm converges, it will only converge to the consensus state as suggested in Eq. (25).

[*Proof of Convergence I*] As discussed in Section 3.2 and Lemma 2, with the proposed distributed optimization algorithm, the problem will converge with finite iteration steps.

[*Proof of Convergence II*] Assume the algorithm will converge to certain state ζ and $\zeta \neq \zeta^*$, where ζ^* is the equilibrium state as suggested in Eq. (25). Since the network topology in this problem is distributed and connected, there must exist agent $j \in N_i$ such that

$$x_i \neq x_j, x_i \in \zeta \quad \text{and} \quad x_j \in \zeta \tag{26}$$

Since ζ is the state of convergence, according to the empirical frequency updating protocol and Definition 4, the empirical frequency for agent i and agent j will be

$$\lim_{k\to\infty} e_i^j(k) = f(x_j, \sigma_j^2) \tag{27}$$

$$\lim_{k\to\infty} e_j^i(k) = f(x_i, \sigma_i^2) \tag{28}$$

Therefore, according to Eq. (23), the new updating process is

$$\begin{cases} x_i' = x_j \\ x_j' = x_i \end{cases} \tag{29}$$

Since $x_i \neq x_j$ the updated state $\zeta' = \{x_1', x_2', \dots, x_n'\}$ is not equal to ζ, according to the definition of convergence as Definition 4, the assumption does not hold. Therefore, if the algorithm converges, it will only converge to the consensus state as Eq. (25). □

4.2. Rendezvous problem for UAVs

A rendezvous problem is a typical multi-agent cooperation problem. For a rendezvous mission, multiple agents are usually required to arrive a rendezvous point through communication with each other under a distributed network topology. A rendezvous problem can be related to various application domain. The authors[30–32] considered the planning of threat-avoiding trajectories for agents for a rendezvous mission in a hostile environment. In this section, without further complicated the discussion, a simple rendezvous problem will be considered and the problem will be handled with the proposed distributed optimization algorithm as discussed in previous sections.

Assume there are n agents located randomly on a 2-dimensioal (2D) plane, and the inter-agent communication is undirected and distributed. Each agent i can only communicate with its neighbour $j \in N_i$. The model for each agent follows the definition of a unicycle model as

$$\begin{cases} \dot{x}_i = V \cos \theta_i \\ \dot{y}_i = V \sin \theta_i \\ \dot{\theta}_i = \omega_i \end{cases} \tag{30}$$

where V is the constant speed for every agent and ω_i is the control input.

The rendezvous problem requires agents to gather to the same location on the 2D plane. The controllers for agents are designed based on their own empirical frequency. The rendezvous mission does not require UAVs arrive the same location at the same time. The inter-UAV communication happens on a predefined time interval, and vehicles will re-adjust their control inputs whenever information update is available. For this rendezvous problem, the global cost function is designed as

$$J_g(\mu_1, \mu_2, \dots, \mu_n) = \sum_{i=1}^{n} J_i(\mu_i, \mu_{-i}) \tag{31}$$

with

$$J_i(\mu_i, \mu_{-i}) = \sum_{j \in N_i} \left[(x_i - x_j)^2 + (y_i - y_j)^2 \right]$$

where $\mu_i = (x_i, y_i, \theta_i)$ is the state of agent i.

Based on the discussion of the distributed optimization algorithm in previous sections, each agent i will establish its local empirical frequency $e_i = \{e_i^1, e_i^2, \ldots, e_i^n\}$ based on the information from its neighbours. Each entry of the empirical frequency e_i^j is the estimation for the state of another agent, i.e. $e_i^j = (e_{ix}^j, e_{iy}^j, e_{i\theta}^j)$ with both e_{ix}^j, e_{iy}^j and $e_{i\theta}^j$ are Gaussian process as $f_{ix}(\mu_j) \sim N(u_{ix,j}, \sigma_{ix,j}^2), f_{iy}(\mu_j) \sim N(u_{iy,j}, \sigma_{iy,j}^2)$ and $f_{i\theta}(\mu_j) \sim N(u_{i\theta,j}, \sigma_{i\theta,j}^2)$, respectively. For a strategy $\mu_i = (x_i, y_i, \theta_i) \in \mathcal{U}_i$, since the global cost function only involves the state of $\{x_j, j \in N_i\}$ and $\{y_j, j \in N_i\}$, the cost of the strategy based on the local empirical frequency is

$$J_i(\mu_i|e_i) = \int \ldots \int J_g(\mu_1, \mu_2, \ldots, \mu_n) f_{ix}(\mu_1) \cdot f_{iy}(\mu_1) \ldots f_{ix}(\mu_n) f_{iy}(\mu_n) \mathrm{d}\mu_1 \ldots \mathrm{d}\mu_n \tag{32}$$

By analyzing $J_g(\mu_1, \mu_2, \ldots, \mu_n)$ and $\{J_i(\mu_i, \mu_{-i}), i \in N\}$ in Eq. (31), the designed problem is consistent with a potential game as defined in Definition 5, and thus, the equilibrium for $\{J_i(\mu_i, \mu_{-i}), i \in N\}$ is consistent with the global objective. Therefore, the cost function for the strategy μ_i of agent i can be simplified as

$$\begin{aligned} J_i(\mu_i|e_i) &= \int \ldots \int J_i(x_i, y_i) f_{ix}(\mu_1) f_{iy}(\mu_1) \ldots f_{ix}(\mu_n) f_{iy}(\mu_n) \mathrm{d}\mu_1 \ldots \mathrm{d}\mu_n \\ &= \sum_{j \in N_i} (u_{ix,j}^2 + \sigma_{ix,j}^2 - 2u_{ix,j} x_i + x_i^2) \\ &\quad + \sum_{j \in N_i} (u_{iy,j}^2 + \sigma_{iy,j}^2 - 2u_{iy,j} y_i + y_i^2) = \Psi_i(x_i, y_i) \end{aligned} \tag{33}$$

With the support of Eq. (33), the optimal control problem is design as

$$J_i = \Psi_i(x_i(t_f), y_i(t_f)) + \int_0^{t_f} \frac{1}{2} r \omega_i^2 \mathrm{d}t$$

$$\text{s.t.} \quad \begin{cases} \dot{x}_i = V \cos \theta_i \\ \dot{y}_i = V \sin \theta_i \\ \dot{\theta} = \omega_i \end{cases} \tag{34}$$

By solving this optimal control problem, the control input and the trajectory for each agent can be obtained for the rendezvous mission.

5. Simulation results

In this section, based on the analysis of the two consensus problems in the previous section, corresponding simulations are conducted to illustrate the theoretical developments.

5.1. Simulation for distributed averaging problem

The simulation for the distributed averaging problem is conducted to illustrate the influence of neighbour numbers and memory size on convergence efficiency. The number of neighbours for each agent is the same. The weights a_{ij} are constant and equal to each other. Every agent can only communicate with its most adjacent agents. The updating protocol adopted in the simulation follows Eq. (15) with the variance values for distribution functions equal 0.

Table 1 and Fig. 1(a) demonstrate the influence of number of neighbours on the algorithm convergence rate. The memory size in this case is infinite, which implies that each agent will record the whole strategy profile for its counterparts.

From the simulation results, it is apparent that the convergence efficiency will increase if the number of neighbours for each agent is increased, which is coherent with the fundamental consensus theory.[33]

Table 1 and Fig. 1(b) demonstrate the influence of memory of the empirical frequency on the algorithm convergence rate. In Table 1, the column "Infinite memory size" represents the case that agents will take all strategies adopted by their counterparts into consideration to make their own decision, while "Memory size 2" and "Memory size 3" columns represent agents will only use the most recent 2 or 3 information to estimate their opponents' strategy patterns.

In order to make the simulation result more clear for demonstration, the information updating process for only one agent is demonstrated in Fig. 1(b). The blue line represents the infinite memory case; the red line represents the case in which the memory size of the empirical frequency equals 2; the green line represents the memory size 3 case.

The result demonstrates that for the network that each agent has less neighbours, the decrease of memory size will improve the performance greatly. However, under network topologies that each agent has more neighbours, there is no clear pattern to describe the influence of memory on the performance of convergence.

5.2. Simulation for rendezvous problem with multiple UAVs

The simulation for the rendezvous problem is conducted to demonstrate the designing procedure of a distributed multi-agent cooperation mission based on the distributed optimization algorithm proposed before. In the simulation, 5 unicycle model based vehicles are considered with constant velocity $V = 10$ m/s. The vehicles are distributed on the 2D plane at points [1 km, 3 km], [3 km, 7 km], [9 km, 10 km], [10 km, 4 km] and [6 km, 2 km].

In this problem, the inter-vehicle collision and obstacle avoidance are not considered for simplicity. The inter-vehicle communication is distributed, and each vehicle can only communicate with its 2 most adjacent neighbours under an undirected network topology. The communication is assumed to happen every 10 s after the mission start, and communication

Table 1 Simulation results for the distributed averaging problem.

Number of neighbours	Infinite memory size	Memory size 3	Memory size 2
3	470	395	306
5	152	128	96
7	66	58	39
10	38	45	19
15	13	23	15

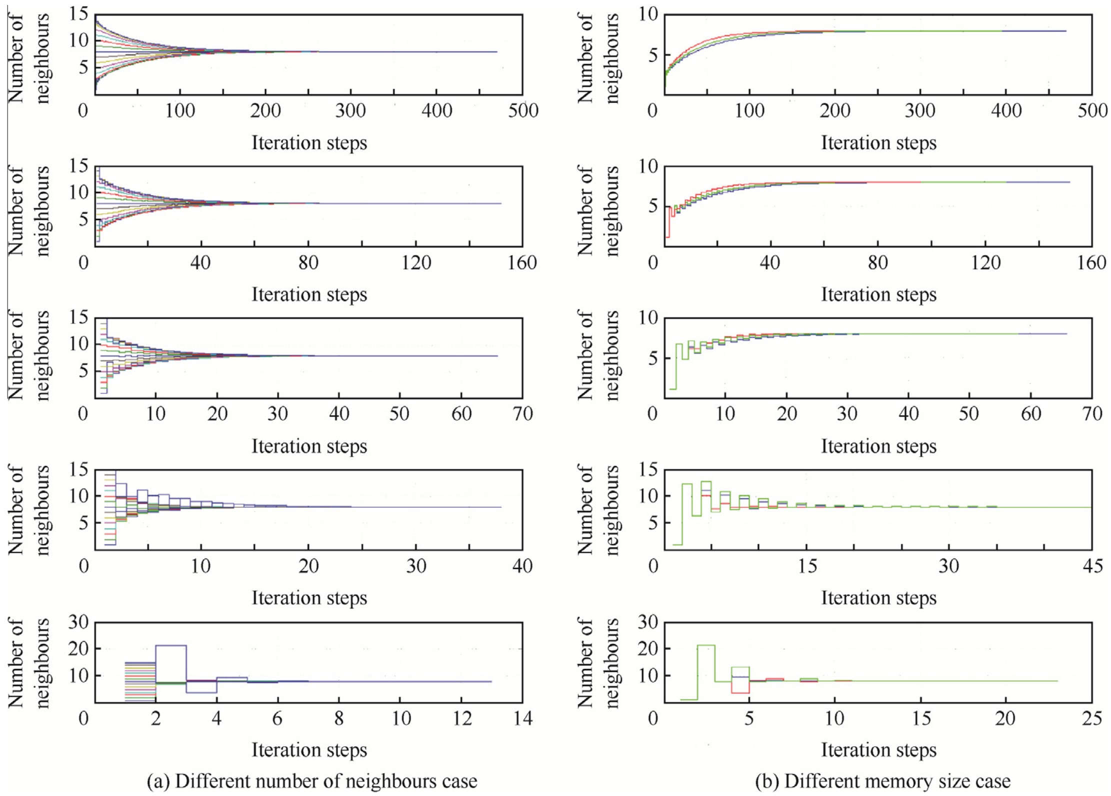


Fig. 1 Simulation for distributed averaging problem.

failure is not considered. The optimization will be terminated when the global cost is smaller than a predefined threshold T, and in the simulation, the threshold is set as $T = 0.1$. The general pseudo-spectral optimization software (GPOPS)[34] is adopted for the optimal control problem.

The trajectories for the vehicles are demonstrated in Fig. 2(a), and corresponding global cost function convergence result with regard to time is illustrated in Fig. 2(b). The convergence of the state x and y for each vehicle are demonstrated in Fig. 3(a) and (b) separately. Fig. 4(a) shows the simulation result of the state θ. The designed controller signal is as demonstrated in Fig. 4(b).

The simulation demonstrates that the designed controller based on the distributed optimization algorithm proposed in the previous section can achieve the rendezvous mission requirement.

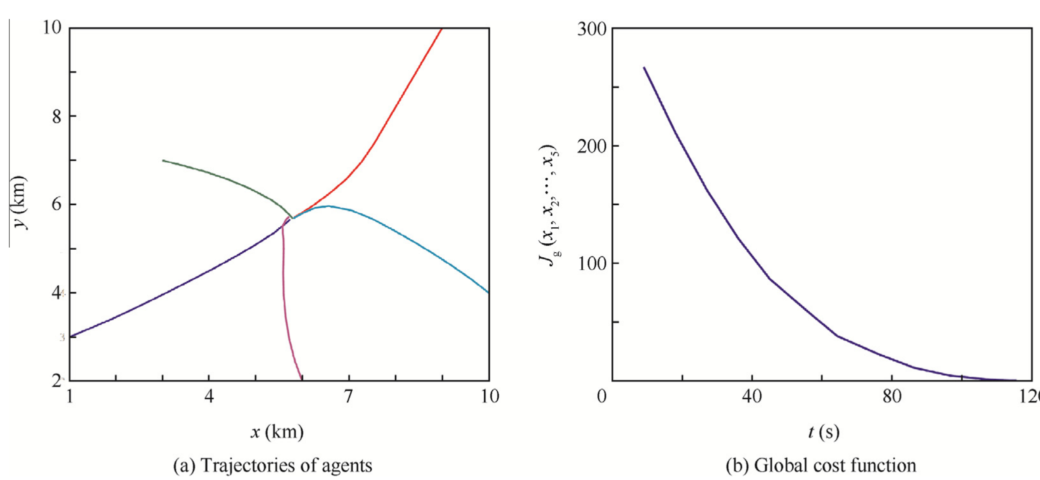
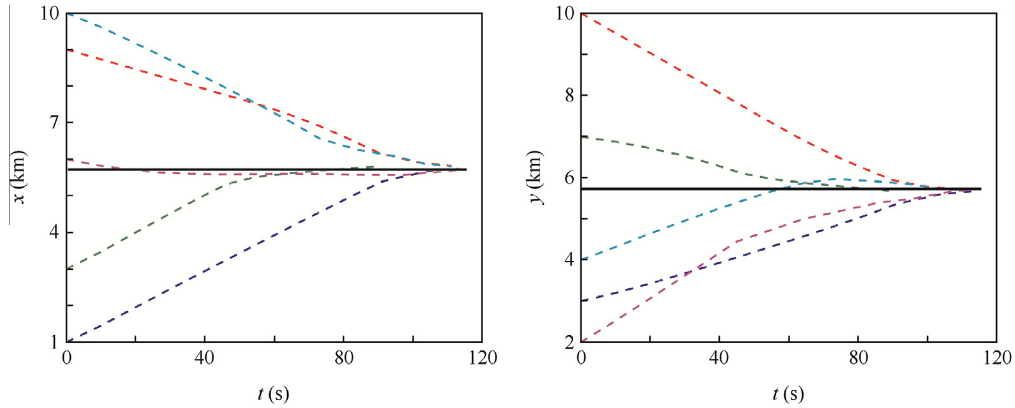
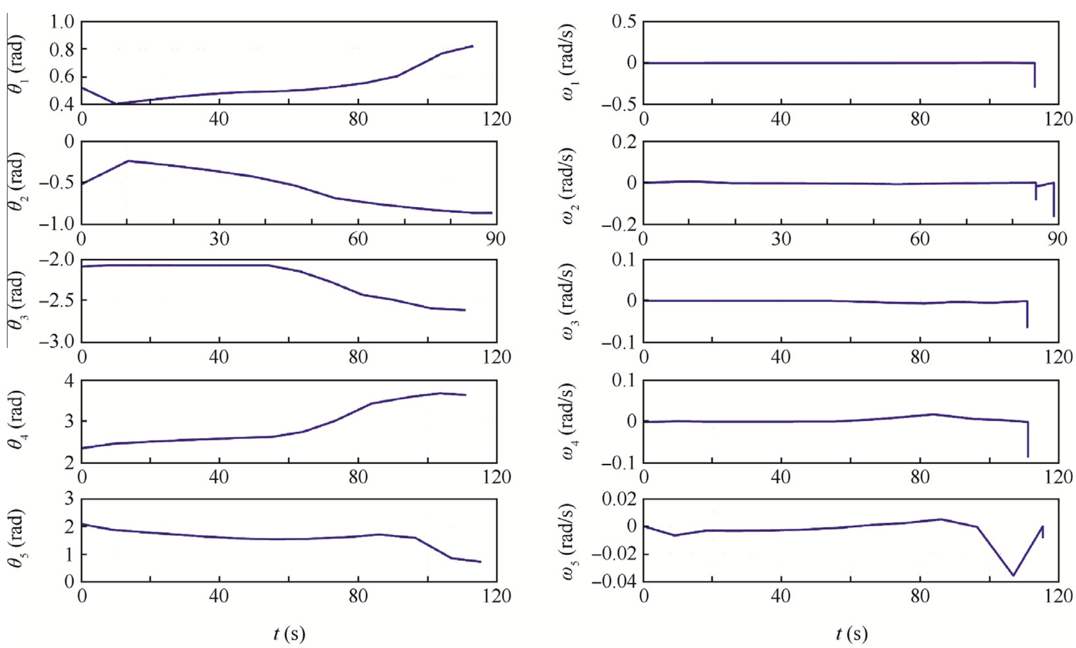


Fig. 2 Simulation results of trajectory for each vehicle and global cost.



(a) Convergence of the state x (b) Convergence of the state y

Fig. 3 Simulation results of states x and y.



(a) State of heading angle (b) Designed control signal

Fig. 4 Simulation results of heading angle θ and control signal ω.

6. Conclusions

(1) A distributed optimization algorithm is designed based on the fictitious play concept. The algorithm assumes agents analyze past observations as if behaviours of their competitors were stationary, and by learning from the strategy patterns of their opponents, agents will decide their own optimal strategies.

(2) The designed distributed optimization algorithm ensures the algorithm autonomy and robustness through the learning behaviour of each agent.

(3) Convergence of the proposed algorithm is proven under game theory framework to guarantee the algorithm will converge to a Nash Equilibrium provided that the original problem design exists one.

(4) By introducing and analyzing the two typical consensus problems with the proposed distributed optimization algorithm, this paper provides a guideline for multi-agent cooperation problem design under game theory framework.

(5) Simulation results are provided to demonstrate that the designed game model for the distributed multi-agent cooperation problems can ensure global cooperation among myopic agents.

Acknowledgements

This work is funded by China Scholarship Council (No. 201206230108) and the Natural Sciences and Engineering

Research Council of Canada Discovery Grant (No. RGPIN 227674).

References

1. Hernández JZ, Ossowski S, Garcıa-Serrano A. Multiagent architectures for intelligent traffic management systems. *Transp Res Part C Emerg* 2002;**10**(5–6):473–506.
2. Abreu B, Botelho L, Cavallaro A, Douxchamps D, Ebrahimi T, Figueiredo P, et al. Video-based multi-agent traffic surveillance system. *Proceedings of the IEEE intelligent vehicles Symposium IV* 2000; Oct 3–5; Dearborn, MI. Piscataway, NJ: IEEE; 2000. p. 457–62.
3. Ren W, Beard RW. *Distributed consensus in multi-vehicle cooperative control: theory and applications*. London: Springer-Verlag; 2008.
4. Ren W, Beard RW. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans Autom Control* 2005;**50**(5):655–61.
5. Remagnino P, Tan T, Baker K. Multi-agent visual surveillance of dynamic scenes. *Image Vis Comput* 1998;**16**(8):529–32.
6. Orwell J, Massey S, Remagnino P, Greenhill D, Jones GA. A multi-agent framework for visual surveillance. *Proceedings of international conference on image analysis and processing*; 1999 Sept 27–29; Venice, Italy. Piscataway, NJ: IEEE; 1999. p. 1104–7.
7. Ivanov YA, Bobick AF. Recognition of multi-agent interaction in video surveillance. *Proceedings of the seventh IEEE international conference on computer vision*; 1999 Sept 20–25; Corfu, Greece. Piscataway, NJ: IEEE; 1999. p. 169–76.
8. Janssen M, de Vries B. The battle of perspectives: a multi-agent model with adaptive responses to climate change. *Ecol Econ* 1998;**26**(1):43–65.
9. Olfati-Saber R. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans Autom Control* 2006;**51**(3):401–20.
10. Jadbabaie A, Lin J, Morse AS. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans Autom Control* 2003;**48**(6):988–1001.
11. Vicsek T, Czirók A, Ben-Jacob E, Cohen I, Shochet O. Novel type of phase transition in a system of self driven particles. *Phys Rev Lett* 1995;**75**(6):1226.
12. Zhang F, Leonard NE. Cooperative filters and control for cooperative exploration. *IEEE Trans Autom Control* 2010;**55**(3): 650–63.
13. Zhang F, Fiorelli E, Leonard NE. Exploring scalar fields using multiple sensor platforms: tracking level curves. *Proceedings of 2007 46th IEEE conference on decision and control*; 2007 Dec 12–14; New Orleans, LA. Piscataway, NJ: IEEE; 2007. p. 3579–84.
14. Saber RO, Murray RM. Consensus protocols for networks of dynamic agents. *Proceedings of the American control conference*; 2003 Jun 4–6; Pasadena, CA. Piscataway, NJ: IEEE; 2003. p. 951–6.
15. Olfati-Saber R, Murray RM. Consensus problems in networks of agents with switching topology and time delays. *IEEE Trans Autom Control* 2004;**49**(9):1520–33.
16. Fax JA. Optimal and cooperative control of vehicle formations [dissertion]. California: California Institute of Technology; 2001 [dissertation].
17. Fax JA, Murray RM. Information flow and cooperative control of vehicle formations. *IEEE Trans Autom Control* 2004;**49**(9):1465–76.
18. Kraus S. Negotiation and cooperation in multi-agent environments. *Artif Intell* 1997;**94**(1–2):79–97.
19. Semsar-Kazerooni E, Khorasani K. Multi-agent team cooperation: a game theory approach. *Automatica* 2009;**45**(10):2205–13.
20. Li N, Marden JR. Designing games for distributed optimization. *Proceedings of 2011 50th IEEE conference on decision and control and European control conference*; 2011 Dec 12–15; Orlando, FL. Piscataway, NJ: IEEE; 2011. p. 2434–40.
21. Li N, Marden JR. Designing games for distributed optimization. *IEEE J Selected Top Signal Process* 2013;**7**(2):230–42.
22. Lin ZJ, Liu HHT. Consensus based on learning game theory. *Proceedings of the 6th IEEE Chinese guidance navigation and control conference*; 2014 Aug 8–10; Yantai, China. Piscataway, NJ: IEEE; 2014.
23. Milgrom P, Roberts J. Adaptive and sophisticated learning in normal form games. *Games Econ Behav* 1991;**3**(1):82–100.
24. Monderer D, Shapley LS. Potential games. *Games Econ Behav* 1996;**14**(1):124–43.
25. Shapley LS. A value for n-person games. 1952 Mar. Report No.: AD604084.
26. Gopalakrishnan R, Nixon S, Marden JR. Stable utility design for distributed resource allocation [Internet]; 2014 [cited 2014 Jun]. Available from: <http://ecee.colorado.edu/marden/files/cdc2014.pdf>.
27. Marden JR, Shamma JE. Game theory and distributed control. *Handbook of game theory, vol. 4*. Forthcoming.
28. Wolfowitz J. Products of indecomposable, aperiodic, stochastic matrices. *Proc Am Math Soc* 1963;**14**(5):733–7.
29. Cao YC, Yu WW, Ren W, Chen GR. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Trans Industr Inf* 2013;**9**(1):427–38.
30. McLain TW, Chandler PR, Rasmussen S, Pachter M. Cooperative control of UAV rendezvous. *Proceedings of the American control conference*; 2001 Jun 25–27; Arlington, VA. Piscataway, NJ: IEEE; 2001. p. 2309–14.
31. McLain TW, Chandler PR, Pachter M. A decomposition strategy for optimal coordination of unmanned air vehicles. *Proceedings of the American control conference*; 2000 Jun; Chicago, Illinois. Piscataway, NJ: IEEE; 2000. p. 369–73.
32. McLain TW, Beard RW. Trajectory planning for coordinated rendezvous of unmanned air vehicles. 2000. Report No.: AIAA-2000-4369.
33. Olfati-Saber R, Fax JA, Murray RM. Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 2007;**95**(1):215–33.
34. Rao AV, Benson DA, Darby C, Patterson MA, Francolin C, Sanders I, et al. Algorithm 902: GPOPS, a MATLAB software for solving multiple-phase optimal control problems using the gauss pseudospectral method. *ACM Trans Math Softw* 2010; **37**(2):22:1–22:39.

Lin Zhongjie received his B.S. degree from the school of electronic information and electrical engineering of Shanghai Jiao Tong University and M.S. degree from the school of aeronautics and astronautics of Shanghai Jiao Tong University. He is currently a Ph.D. candidate at the University of Toronto Institute for Aerospace Studies (UTIAS). His main research interests are multi-agent cooperation problems, distributed optimization, game theory and task assignment domain.

Liu Hugh hong-tao is a professor at the University of Toronto Institute for Aerospace Studies (UTIAS), Toronto, Canada, where he also serves as the Associate Director, Graduate Studies. His research work over the past several years has included a number of aircraft systems and control related areas, and he leads the ‘‘Flight Systems and Control’’ (FSC) Research Laboratory.