2020 International Conference on Unmanned Aircraft Systems (ICUAS)
Athens, Greece. September 1-4, 2020

Cooperative Game Theory based Multi-UAV Consensus-based Formation Control

Liwei Jiang[1], Felipe Gonzalez[2], Aaron McFadyen[2]

Abstract— **This paper presents a cooperative differential game theory approach to designing a consensus-based formation control strategy for multi-UAVs (agents). The consensus-based formation control problem is defined as a tracking task where the states of each agent must follow a given reference. Position states are given as fixed offsets between each agent to achieve a formation, whilst velocity and attitude should be identical amongst agents to archive consensus. Using a cooperative game theory based approach, each agent acts to reduce the weighted team cost to reach consensus. The control strategy design is based on open-loop design, closed-loop design, and a semi-open-loop design concept to better account for different hardware configurations. The control approach is validated via simulations to show how cooperative game theory based consensus formation control can outperform non-cooperative game theory control and general optimal control with respect to reducing formation deviation. The semi-open loop design is proven to be suitable for practical applications.**

I. INTRODUCTION

Multi-drone systems offer an attractive solution to tasks like traffic patrol, search and rescue, aerial mapping, fire-fighting etc. Such tasks may require that each agent acts independently or as group to achieve a common goal. Some applications may require the agents to complete a task in a certain time whilst others may require specific motion patterns. In any case, autonomous solutions to the multi-agent control must address a range of challenging problems when practical constraints are considered.

Multi-agent system architectures can therefore vary as much as the level of intelligence required by each drone. Some multi-agent systems seek to achieve consensus, which is loosely defined to occur when all agents act in the same way. Amongst other challenges, agent topology constraints and associated communication delay is two of the key factors affecting the stability of such systems. The philosophical solution to avoid communication delay is to not communicate, in which case game theory provides some possible solutions. In the classic example of game theory, prisoner's dilemma, two suspects who can't communicate must make their own choice under the same rules to maximize their own interests. Through rule design, a Nash equilibrium will be reached. On the contrary, if they try to maximize a team-level goal via communication there may be multiple equally optimal outcomes, which is called cooperative game theory.

This paper considers multiple drones as players participating in such a cooperative game. The basic idea behind cooperative game theory including how to find the Nash equilibrium point of multi-agent in cooperative differential games is described in [2]. The mathematical basis for finding Pareto's optimality at multiple Nash equilibrium points is discussed in [1] and [8]. The basic framework for dynamic modeling of the agent behaviour comes from [5]. Agents are predictable and rational to a certain extent, because their behavior can be described by differential equations in a certain linear domain. Control inputs are the decisions agents need to make where the cost equation used in general optimal control determines individual payoff. Therefore, a differential game is formed. Through the design of rules and explicit consideration of a practical system model and real environment, this paper offers the following contributions:

- Implementation of cooperative differential game theory approach in consensus-based formation control including controller optimization.
- Proposal and analysis of semi-open loop design to make this approach more useful in practical cases.

The rest of this paper is organized as follows. Section II gives mathematical formulation of the problem and proposed solution. Simulation results including optimization strategies for control design are presented in Section III. Concluding remarks are offered in Section IV.

II. PROBLEM FORMATION AND SOLUTION

A. Preliminaries and Notation

Before our problem is formed, the consensus control needs to be defined mathematically. According to [12], two mainstream definitions of consensus control are adopted in this paper: In a system with N agents, if the agent state x can satisfy (1), we have *absolute consensus*, and if it can satisfy (2), we have *relative consensus*. The control that allows the system in any initial state to reach consensus is called consensus control.

$$\text{Absolute Consensus}: \quad \forall i,j \in N : \lim_{t\to\infty} \left\| \mathbf{x}_i - \mathbf{x}_j \right\| = 0 \tag{1}$$

$$\text{Relative Consensus}: \quad \forall i,j \in N : \lim_{t\to\infty} \left\| \mathbf{x}_i - \mathbf{x}_j - \mathbf{d}_{ij} \right\| = 0 \tag{2}$$

where $\mathbf{d}_{ij}$ is the desired offset between the states of agent i and j. Consider a system consists of N drones. Our mission is to complete consensus-based formation flight, that is, to complete a prescribed trajectory while maintaining a certain distance between each agent to achieve relative consensus whilst all agents simultaneously hold the same attitude and velocity to archive absolute consensus.

[2] are with Science and Engineering Faculty, Electrical Engineering & Robotics, Queensland University of Technology
[1] with Institute of Flight Mechanics and Controls, University of Stuttgart
edwardaffed@gmail.com

978-1-7281-4277-7/20/$31.00 ©2020 IEEE

This work adopts the standard aviation notation. The inertial reference frame $\mathbf{I}$ and the body frame $\mathbf{B}$ are defined where $\mathbf{I} : \mathbf{r} = (x, y, z)$ defines UAV position $\mathbf{r}$. The UAV attitude $\Omega = (\phi, \theta, \psi)$ (roll,pitch,yaw) parameterises the Euler angle rotation matrix to facilitate $\mathbf{B}$ to $\mathbf{I}$. Angular velocity ω is defined such that $\omega = (p, q, r) = (\dot{\phi}, \dot{\theta}, \dot{\psi})$.

B. Dynamic Model and Graph Theory

According to [2][3], the dynamics of typical low-level attitude controllers (inner loop) in drones can be described by a set of first-order equations such that the behavior of all drones are assumed to be identical. Furthermore, small angle theory is assumed to hold and each drone is only subject to gravity and thrust. Based on these assumptions, a simplified dynamic model for each drone linking high level commands (outer loop) to states can be found via linearisation at an equilibrium hover position. Drone dynamics can then be decoupled into four dimensions concerning x-axis behavior, y-axis behavior, altitude behavior and heading (yaw rate) behavior. Considering real hardware constraints, the pitch angle, roll angle, thrust, and yaw rate commands constitute the inputs to control the drone motion in the above four dimensions respectively. The system state equation for each drone are defined using state space notation such that

$$\dot{\mathbf{x}}_i(t) = A_i \mathbf{x}_i(t) + b_i u_i(t) \tag{3}$$

$$where: \ A_i = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a_\theta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -a_\phi & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -a_r \end{pmatrix}, b_i = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a_\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a_\phi & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{m} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a_r \end{pmatrix}$$

and $m = 0.027\,kg$ is the drone mass, $g = 9.81 \ m/s^2$ is gravitational acceleration and the empirically derived constants a_θ, a_ϕ, a_r are the coefficients in a set of first-order equations [5] obtained via system identification similar. The system state vector $\mathbf{x}_i = (x, \dot{x}, \theta, y, \dot{y}, \phi, z, \dot{z}, \psi, r)^T \in \mathbb{R}^{10\times 1}$ and control input vector $u_i = (\theta^d, \phi^d, \delta_F^d, r^d)^T \in \mathbb{R}^{4\times 1}$ are defined where i indicates that the parameter belongs to the i^{th} drone and d denotes the demanded (or desired) command from the outer loop controller. The dynamic behavior of the entire system (of N drones) in each dimension can be described by an equation of the form

$$\dot{\mathbf{x}} = A\mathbf{x}(t) + \sum_{i=1}^{N} \{B_i u_i(t)\} \tag{4}$$

where $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T)^T$, $A = E_N \otimes A_i$, E_N is the N-dimensional identity matrix and $B_i = (0 \ldots 0\ 1\ 0 \ldots 0) \otimes b_i$ where the 1 corresponds to the i^{th} term in b_i.

In order to build and minimize the cost equations of multi-drone system, formation deviation and energy consumption are given by quadratic expressions of the form

$$\begin{aligned} J_i = & \left(\left(\mathbf{x}_i(t_T) - \mathbf{x}_i{}^d(t_T) \right)^T Q_{i,T} \left(\mathbf{x}_i(t_T) - \mathbf{x}_i{}^d(t_T) \right) \right) \\ & + \int_{t_0}^{t_T} \left(\left(\mathbf{x}_i(\tau) - \mathbf{x}_i{}^d(\tau) \right)^T Q_i \left(\mathbf{x}_i(\tau) - \mathbf{x}_i{}^d(\tau) \right) \right) d\tau \\ & + \int_{t_0}^{t_T} \left(u_i^T(\tau) R_i u_i(\tau) \right) d\tau \end{aligned} \tag{5}$$

where R_i and $Q_i = \hat{D}\hat{W}_i\hat{D}^T$ are the control and state weighting matrices assuming the same meaning as it in general optimal control. Here, $\hat{D} = D \otimes E_n$, $\hat{W}_i = W_i \otimes E_n$ with $W_i = diag\{w_{i,j}\}$ where E_n is the n-dimensional identity matrix and n is the length of the state of single drone. The network G gives the incidence matrix D to simulate the communication between N agents. In a network graph G, N vertices represent N UAVs in the system, and edges represent the existence of the message transmission between UAVs. For any UAV i and UAV j, $D_{ij} = 1$, if i receives messages from neighbor j, $D_{ij} = -1$, if i sends messages to neighbor j and $D_{ij} = 0$, if i and j are not neighbors and thus do not communicate. The degree of importance of information on different edges connected to agent i is given by W_i where $W_i = diag\{w_{ij}\}$. $w_{ij} > 0$ means that information in this edge is important and $w_{ij} = 0$ means it is not. For all existing communications, the weight w_{ij} is set to 5.

The subscript T indicates an instant of this parameter at the final time $t = t_T$. For example, $Q_{i,T}$ represents the value of Q_i at $t = t_T$. But usually Q_i can be considered as a time constant. Additionally, to ensure the leader tracks the target trajectory $Q_{i,l}$ of the leader agent has the following settings

$$\begin{aligned} Q_{i,l} &= Q_i + diag\{0, \ldots, 0, q_i, 0, \ldots, 0\}, \\ q_i &= q \cdot E_N \end{aligned} \tag{6}$$

with q_i at i^{th} term and q is the weight of trajectory tracking task for the leader. Note, Q_i can be proven to be real symmetric positive semidefinite and $Q_i = 0$ if leader is expected to give up tracking formations. The state vector $\mathbf{x}_i \in X$, and X also needs to satisfy the convexity condition that ensures X is a compact and convex subset of the $\mathbb{R}^n$ which contains $\mathbf{x}_i{}^d$ in itself.

C. Game Design

The design of the macro game rule can consider either non-cooperative games or cooperative games. For non-cooperative games, there is a Nash equilibrium point and any player will have a dominant strategy $u_i{}^*$ such that

$$J_i(u_1{}^*, u_2{}^*, \ldots u_i{}^*, \ldots, u_N{}^*) \leq J_i(u_1, u_2, \ldots, u_i{}^*, \ldots, u_N), \tag{7}$$

where $i = 1 \ldots N$. For cooperative games, all players have a public agreement that any decision they make must not cause losses to others. This means that the choice of any player alters the overall (global) cost and there exists a set of actions for all players that results in equally optimal outcomes such that Pareto's Optimality is satisfied. The mathematical condition is that the following inequality holds for N players

$$\mathbf{u}^* \in \underset{\mathbf{u}\in U}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \{\alpha_i J_i(\mathbf{u})\} \right\} \tag{8}$$

where $\alpha_i \in [0,1]$ and $\sum_{i=1}^{N} \alpha_i = 1$. The object of minimization is a weighted summation term where α_i is the weight coefficient representing the proportion of the cost of a certain drone in the total cost. The weight of the leader can be set higher, because its core task is to follow the target path. Of note, over-reliance on leader can lead to an increase in formation deviation, hence this will be an optimized as discussed later.

According to the closed loop design provided by [2][7][8], for a system that satisfies the agent behavior equation (3) and system behavior (4), there exists a set of equations $s(t)$ and $p(t)$ that can be used to construct a set of input strategies $u_i, i = 1 \ldots N$ that can satisfy Pareto's Optimality inequality (8) such that

$$u_i(t) = -\frac{1}{\alpha_i} R_i{}^{-1} B_i{}^T (s(t)\cdot \mathbf{x}(t) + p(t)) \tag{9a}$$

$$\dot{s}(t) = -sA - A^T s - \sum_{i=1}^{N} \{\alpha_i Q_i\} + s \cdot \sum_{i=1}^{N} \left\{ \frac{1}{\alpha_i} B_i R_i{}^{-1} B_i{}^T \right\} \cdot s \tag{9b}$$

$$with\ s(t_T) = \sum_{i=1}^{N} \{\alpha_i Q_{i,T}\}$$

$$\dot{p}(t) = \sum_{i=1}^{N} \{\alpha_i Q_i\} \cdot \mathbf{x}^d + s \cdot \sum_{i=1}^{N} \left\{ \frac{1}{\alpha_i} B_i R_i{}^{-1} B_i{}^T \right\} \cdot p \tag{9c}$$

$$with\ p(t_T) = -\sum_{i=1}^{N} \{\alpha_i Q_{i,T}\} \cdot \mathbf{x}^d$$

where $s(t) \to s$ is essentially the feedback coefficient of the system's own state and $p(t) \to p$ as it contains fixed information about the design path. The numerical solution of the variable differential equations above requires a large amount of computational capability so it is common for pseudo-distributed systems to utilise a central platform for computation. The calculation of the closed-loop controller will thus be demanding, and the control loop may not be stabilisable due to the discretisation. For these reasons, open-loop control can be applied via replacement of closed-loop state feedback with estimation based on initial conditions and the model where

$$u_i(t) = -\frac{1}{\alpha_i} R_i{}^{-1} B_i{}^T (s(t)\cdot \mathbf{x}(t) + p(t)) \tag{10a}$$

$$\dot{s}(t) = -sA - A^T s - \sum_{i=1}^{N} \{\alpha_i Q_i\} + s \cdot \sum_{i=1}^{N} \left\{ \frac{1}{\alpha_i} B_i R_i{}^{-1} B_i{}^T \right\} \cdot s, \tag{10b}$$

$$with\ s(t_T) = \sum_{i=1}^{N} \{\alpha_i Q_{iT}\}$$

$$\dot{p}(t) = \sum_{i=1}^{N} \{\alpha_i Q_i\} \cdot x^d + s \cdot \sum_{i=1}^{N} \left\{ \frac{1}{\alpha_i} B_i R_i{}^{-1} B_i{}^T \right\} \cdot p, \tag{10c}$$

$$with\ p(t_T) = -\sum_{i=1}^{N} \{\alpha_i Q_{iT}\} \cdot x^d(t_T)$$

$$\mathbf{x}(t) \cong c(t,0)\,\mathbf{x}_0 + e(t) \tag{10d}$$

$$\dot{c}(t,0) = \left(A - \sum_{i=1}^{N} \left\{ \frac{1}{\alpha_i} B_i R_i{}^{-1} B_i{}^T \right\} \cdot s \right) \cdot c, \tag{10e}$$

$$with\ c(t_0,0) = I$$

$$\dot{e}(t) = \left(A - \sum_{i=1}^{N} \left\{ \frac{1}{\alpha_i} B_i R_i{}^{-1} B_i{}^T \right\} \cdot s \right) \cdot e - \sum_{i=1}^{N} \left\{ \frac{1}{\alpha_i} B_i R_i{}^{-1} B_i{}^T \right\} \cdot p, \tag{10f}$$

$$with\ e(0) = 0$$

It should be noted that the coefficient function of the initial condition is abbreviated as c, and the remaining term representing the effect of external input on the system state is abbreviated as e. As the main focus of this paper is to analyze the optimization of this control approach, the proof of this solution can be found in [8]-[10] and is based on Pontryagin's Maximum Principle (PMP) for closed-loop design and the Hamilton–Jacobi–Bellman equation (HJBE) [1] for open-loop design.

III. SIMULATION AND OPTIMIZATION

A. Simulation Setup

Simulation parameters that require initialisation include the specified trajectory, team structure, drone states and performance metrics. All simulations are executed using MATLAB r2019a.

The specified team structure consists of $N = 5$ drones with leader defined by $i = 1$. The formation is expressed by the distance from the drone to the central trajectory as given in Table I. The communication topology is shown in Fig. 1 along with the incidence matrix D required to set the weight matrices W_i. We set the weight of the leader's task from 0 to 1 if tracking the formation. The weight of leader's task w.r.t. trajectory tracking $q = 5E_N$ is set such that $diag\{0, 0, \ldots, q, \ldots 0\}$. Lastly, the team payoff function weights are given by α=[0.2,0.2,0.2,0.2,0.2]. It can be seen that initially the payoff of each member, including the leader, is equally important in the entire team payoff. This weight coefficient will be changed later to study the influence of leadership on the team.

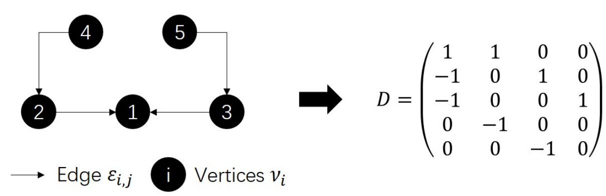

Fig. 1. Communication Network

The specified trajectory is defined via a set of desired waypoints denoting a central path for the drones to track (see Table IV. The trajectory is also parameterised by a mean velocity $\bar{v} = 0.2$ ms^{-1} and minimum turning radius radius $r_{min} = 0.15$ m. Further details on the associated path planning algorithm used to construct a smooth trajectory between waypoints under such turning constraints can be found in the Appendix.

TABLE I

DESIRED FORMATION

Drone	1	2	3	4	5
distance offset x	1	-1	3	-3	-4
distance offset y	4	-2	1	-3	-4
distance offset z	3	1	-2	-1	-4

TABLE II

COMMUNICATION WEIGHT MATRICES: $W_i = diag\{w_{i,edge}\}$

Edges	$w_{1,edge}$	$w_{2,edge}$	$w_{3,edge}$	$w_{4,edge}$	$w_{5,edge}$
(1,2)	5	5	0	0	0
(1,3)	5	0	5	0	0
(2,4)	0	5	0	5	0
(3,5)	0	0	5	0	5

TABLE III

DESIRED WAYPOINTS

WayPoints	1	2	3	4	5	6	7	8	9	10
x	0	1	1	2	2	3.5	3.5	2.5	3	2
y	0	0	1	1	0	0.5	1.5	2.5	3.5	4
z	1	1	2	1	1	1	3.5	3.5	2.5	1.5

The drone states require initialisation in such a way that the linearisation assumptions underpinning the model are not severely compromised. The number of drone's states effecting x-axis after decoupling is 3 and include the x-axis displacement x in meter, x-axis speed $\dot{x}$ in meters per second, and the acceptable input pitch angle θ (in rad). A similar set of parameters for the remaining decoupled systems results and the initial drone states for the 5 drone configuration are given in Table IV.

TABLE IV

INITIAL CONDITIONS

Initial states	x_0	$\dot{x}_0$	θ_0	y_0	$\dot{y}_0$	ϕ_0	z_0	$\dot{z}_0$	ψ_0	r_0
Drone.1	-1	-1	0	0	0	0	1	-1	1	0
Drone.2	1	0	0	2	-1	0	3	0	0.5	1
Drone.3	0	3	3	4	-2	0	0	3	1	1
Drone.4	0	0	0	0	0	0	0	0	0	0
Drone.5	-1	1	1	-1	1	1	2	0	-0.5	0

Remark: *The control strategy (9a) is formed in continuous time, but will be solved via numerical methods which means that the control input has been discretized. The process of designing a controller using this approach is called emulation. Controller design by emulation will be directly conducted using a continuous time model and then the control signal will be sampled according to hardware working cycle.*

For the evaluation of a tracking task, performance metrics are required. After the task is initialized, there must be a deviation between the actual initial position of the agent and the planned initial position. If the performance of this part of the controller is directly judged by the tracking deviation in the general sense, the obtained results will be dominated by the initial conditions. Hence this part of the evaluation does not have a generalized effect and will be ignored directly in the calculation. The second part of the tracking task is

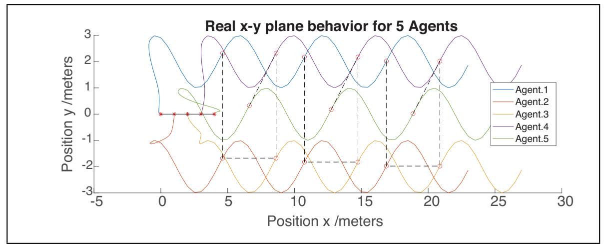

Fig. 2. Agent 1 - 5 $x-y$ position under formation control

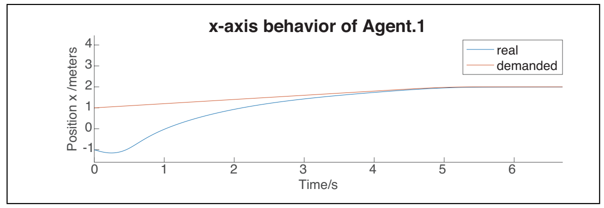

Fig. 3. Agent 1 x position during mission start

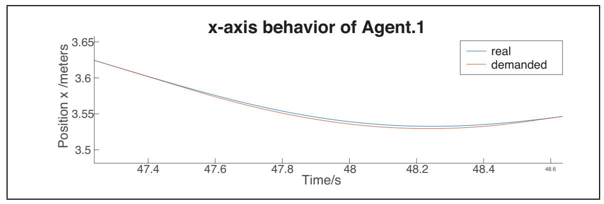

Fig. 4. Agent 1 x position during sharp turn.

tracking the target path, which is the core phase of tracking. The performance at this stage can be characterized by the average of the position tracking deviation at discrete time points, called the time average tracking deviation (TATD) and defines such that

$$TATD = \sqrt{\frac{1}{N_D} * \frac{1}{N_T} \sum_{i=1}^{N_D} \sum_{t=t_0}^{N_T} \left(\mathbf{x}_{i,t} - \mathbf{x}_{i,t}^d\right)^T \left(\mathbf{x}_{i,t} - \mathbf{x}_{i,t}^d\right)} \quad (11)$$

where $N_D = 5$ is the number of involved agents and N_T is the number of discrete time steps due to numerical computation and differences in the path manager processing times. With this performance parameter, different control strategies can be designed then optimized for practical scenarios. It can be seen in (9a) and (10a) that there are two influencing factors that can be optimized according to the actual situation and include the weight R_i of the control input and the team payoff weight α_i. Combined with the computational capability of the real platform host, the optimization of a semi-open loop design is also proposed.

B. Results: Unoptimized Closed-loop Deign

First the overall performance is demonstrated for closed loop control of the formation without parameter optimisation in Fig. 2. In this game, the team penalty caused by the formation error is high so the formation is very stable at the cost of degraded trajectory tracking performance. These results are provided as an example and we vary the demanded path for all other analysis. Additionally, we will focus only on the x-behavior for brevity.

From results shown in Fig. 3-4, the tracking deviation is obviously mainly determined by the initial deviation so including the beginning part of the trajectory when evaluating

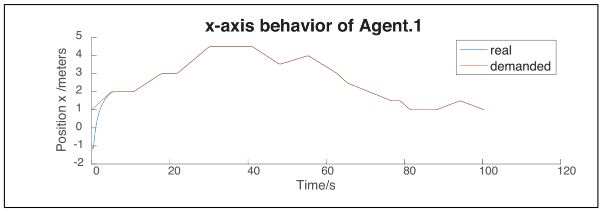

Fig. 5. Agent 1 actual and demanded x- position for the entire mission

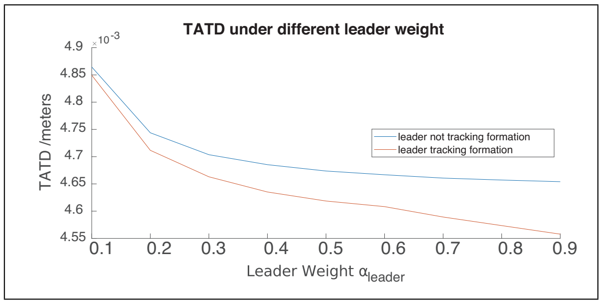

Fig. 6. Time Average Tracking Deviation for different leadership

TABLE V

TIME AVERAGE TRACKING DEVIATION FOR DIFFERENT LEADER

Leader	1	2	3	4	5
TATD($\times 10^{-3}$)	3.891	3.959	358.8	3.956	4.013

controller performance can be misleading. Therefore, the TATD of all tasks in this simulation is calculated from 5 seconds after the start of the task. During the trajectory tracking stage, path deviation is mainly concentrated at the sharp turns in the path. For results shown in Fig. 2 - 5 links to related videos can be found in the Appendix respectively.

C. Results: Optimized Design

We consider optimising the controller design. For this analysis, only one dimension will considered which is the x-axis behavior with all system parameters set to be the same as mentioned before. The logical path for the optimisation and associated analysis is

1) Based on the closed-loop control design, change the team's payoff weight α_i to find an optimization strategy regarding the payoff weight (**Optimization 1**).
2) Based on the open-loop control design, change the control input weight R_i to find an optimization strategy regarding the control input weight in payoff (**Optimization 2**).
3) Based on the actual computing power, change the length of the open-loop time period T_s and find an optimization strategy regarding semi-open loop (**Optimization 3**).

Optimization 1: Team Payoff Weight α_i

The influence of leadership weight and leader task on team formation performance will be analyzed. The leader task can be of two types: simultaneously tracking a required formation and target path or tracking a target path only.

Results depicting the impact of changing the team payoff weight on TATD are shown in Fig. 6 and Table V. It can be concluded that

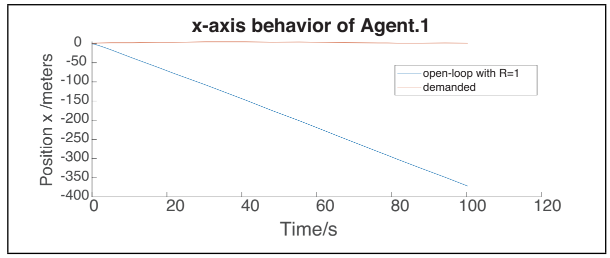

Fig. 7. Open-loop design with inclination protection

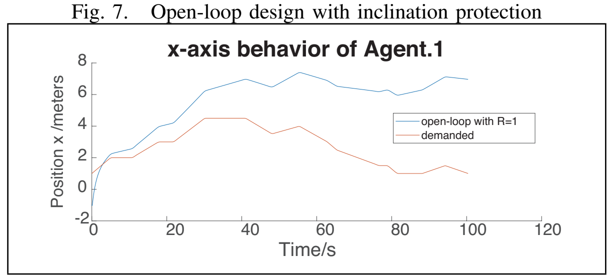

Fig. 8. Open-loop design without inclination protection

- When the system is stable, TATD of cooperative game theory based control approach is always lower than that of the general optimal control, which leads to TATD = 0.0058 meters,
- If the leader tracks the target path as well as the demanded formation, the TATD is always relatively smaller than case of tracking the path only,
- As the leader weight increases in the team, TATD gradually decreases, but when the leader weight is equal to 100%, the system is completely unstable,
- When all drones have the same weight, Table V shows that the exchange of leaders has no obvious effect, except when the leader is drone No.3, which causes 100 times TATD compared to the others. As such, the impact of changing leader requires further study, before which, drone No.1 is set as the leader in all simulations.

Optimization 2: Control Input Weight R_i

Before further analysis is presented, we defined the concept of mandatory inclination protection. In order to ensure the establishment of the small angle theory, the set value of the control output will be limited to a certain range. In this paper, the absolute values of the pitch angle and roll angle are limited to $\pi/4$, yaw rate is limited to 2π rads^{-1} and the thrust is limited according to the actual power of the drone[1]. In open-loop, the calculation of drone control strategies relies on position estimation and the initial conditions. The premise of an exact position estimate is based on no unknown mandatory inclination protection. However, even with known inclination protection, open-loop designs have shown visible tracking deviation because the controller will gradually fail as the estimation error is integrated. For example, as shown in the Fig. 7 and 8, when the control input weight $R_i = 1$ for all drones the controller diverges with inclination protection and converges without inclination protection.

Mandatory inclination protection is necessary because it is the basic premise of model linearisation. As the control

[1]Note that this protection always exists in all of our design.

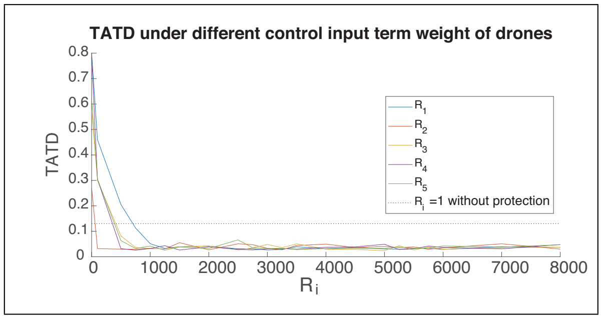

Fig. 9. Time Average Tracking Deviation for different input weights

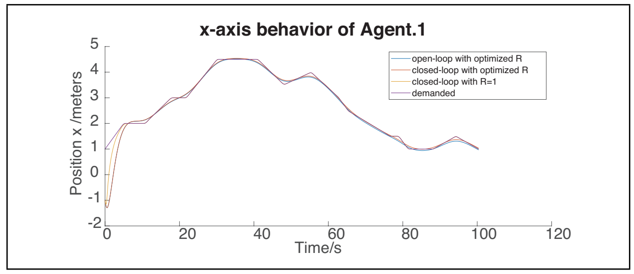

Fig. 10. Optimized open-loop design vs closed-loop design under same R_i

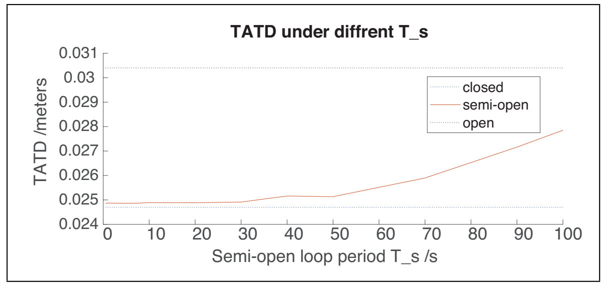

Fig. 11. Optimization of the semi-open loop period

input weighting has not yet been optimised, the control input has not been well tracked. If this weight can be increased, fewer control inputs can achieve the control objective and we sacrifice some tracking accuracy. As such, we explored the option of changing the weight R_i for the control input term for different drones in the team payoff and then designed different open loop controllers accordingly. Additionally, since the range of the control input weight R_i for different drones can lead to different minimum field of TATD, a sensitivity analysis was also conducted.

TABLE VI
R_i SENSITIVITY ANALYSIS USING MINIMUM FIELD OF TATD

R_1 sensitivity	5580	5590	5600	5610	5620
TATD($\times 10^{-3}$)	40.8	39.4	**25.6**	30.2	31.4
R_2 sensitivity	985	995	1005	1015	1025
TATD($\times 10^{-3}$)	32.8	34.3	**27.4**	39.7	40.5
R_3 sensitivity	1000	1025	1050	1075	1100
TATD($\times 10^{-3}$)	39.3	40.2	**25.3**	25.7	41.3
R_4 sensitivity	1000	1025	1050	1075	1100
TATD($\times 10^{-3}$)	39.3	28.8	**27.7**	31.8	28.5
R_5 sensitivity	2900	2910	2920	2930	2940
TATD($\times 10^{-3}$)	39.3	35.1	**30.4**	31.9	44.3

Results depicting the impact of changing control weighting on TATD are shown in Fig. 9 and Table VI. It can be concluded that

- The range of the control input weight R_i that causes the minimum value of the team formation error is different for different drones.
- When the weight R_i of any drone is less than 500, it will cause the system to diverge.
- There is no need to increase the weight without limit.
- If the control input weight of the entire team exceeds a certain range, it will also cause the formation tracking failure.

In view of the above considerations, the optimization regarding control input weights were selected such that $[R_1, R_2, R_3, R_4, R_5] = [5600, 1005, 1050, 1050, 2920]$. Under such optimization, the result of the open-loop design with inclination protection is shown in the Fig. 10 with links to related videos found in Appendix.

Compared with the closed-loop design with $R_i = 1$ in Fig. 4, the TATD of the open-loop design is increased about 5 times. This shows that the error brought by the open loop accumulates with time and added weight of R_i will also reduce formation performance. Therefore, a semi-open loop design is proposed in this work, that is, on the basis of the traditional open loop, at regular intervals, the drone is allowed to read the information of all members of the team and calculate the control input for the remaining tasks based on the update. This time interval T_s is called a semi-open loop period. If T_s is infinitely small, it is closed-loop control. On the contrary, a large value will approximate a purely open-loop control approach. We find a suitable T_s for the host platform hardware via a practical optimization method. One should note that there are fundamental differences between this design and the situation where the task is performed in stages. First, each control strategy update is re-planned for all remaining tasks. Second, the semi-open-loop design uses multi-threaded tasks in the software architecture. The update of the control strategy and the sending of the control command are performed simultaneously.

Remark *The main contribution of this paper is on parameter optimization and the proposal of semi-open loop design. This method is an emulation method, about which one still has limited analytical means to prove stability or to calculate the stability margin. Therefore, stability of our approach is shown through numerical calculations.Proofs surrounding open and closed loop control designs are provided by [2][7][8][10].*

Optimization 3: Semi-open Loop Control (Period T_s)

As expected, Fig.11 shows that a shorter T_s leads to a smaller TATD. The TATD lower limit results from the closed-loop design whilst the upper limit is not surprisingly close to the open loop result. Additionally, the time for the platform to calculate the new control input is about 2 seconds. Considering a series of hardware delays (i.e. actual communication), the semi-open loop period can be set within the interval of 2 to 3 seconds. Even if extending the semi-open-loop period to 10 seconds, compared with closed-loop design, TATD has only increased by less than

2%. This shows that the tolerance of the semi-open-loop control to communication and measurement delay is very high and thus better suited to real situations with such constraints on communication delays.

Remark: *This work does not address the optimization of communication topology D and associated weight matrix W. Our tests have shown (a) different topologies D require different weight matrices W to satisfy system stability and (b) the direction of information transmission does not affect formation deviation when inter-drone communication is restricted to only the nearest/immediate neighbour.*

IV. CONCLUSIONS

This paper presented a cooperative game-theory approach to consensus-based formation control for multi-UAV systems. Investigations were conducted on the impact of different control architectures and associated design parameters in the context of practical systems in order to capture expected behaviour under real constraints such as communication delay. This work draws the following conclusions, noting work is required to completely characterise the performance of the proposed approaches:

- Consensus formation control based on cooperative differential game theory has been proven to be significantly better than that of non-cooperative game theory and general optimal control by comparing the time-averaged tracking deviation.
- For a team with leaders, changing leadership can affect formation errors depending on the network symmetry.
- For effective open-loop design, the weight of the control input needs to be added to the cost equation to ensure that unknown mandatory inclination protection will not be trigged. But the formation error of optimized open-loop design is always greater than closed-loop design.
- With a certain semi-open-loop period, the performance of controller is only 2% lower than that of closed-loop design, but T_s can be longer than 10 seconds.

Further research included investigating suitable analytical methods to optimize communication topology via switching, characterise the impact of leadership transitions and better manage communication delays or mismatch.

APPENDIX

A. Related Videos

Related video files can be found at `https://youtu.be/vLI3iHY4y-w` (x-axis behavior of 5 Drones), `https://youtu.be/votVxR1Wqvg` (x-axis behavior of Drone No.1 in details), `https://youtu.be/1XZwa0sX_bA` (x-y demonstration of 5 Drones), `https://youtu.be/gWdGL-kwErw` (comparison of optimized open-loop / closed-loop designs)

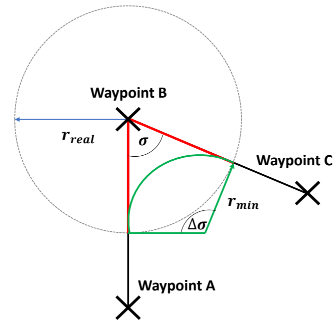

Fig. 12. Real Turn radius to replace sharp corners in demanded path

B. Path Manager

According to [5], a path manager first connects specified points via stringing together linear line segments between the given points in 3D space. At each waypoint a coordinate transformation is then performed such that the first trajectory segment represents the new x-direction, and the new z-direction is perpendicular to the plane spanned by the two segments associated to each waypoint. The new y-direction complements the system to a right-handed trihedron.
In the new coordinates, based on a predefined minimal turn radius and the angle between two consecutive path segments at a waypoint, a real turn radius to the waypoint can be calculated using

$$r_{real} = r_{min} \cdot \tan \left| \frac{\Delta\sigma}{2} \right| = r_{min} \cdot \tan \left| \frac{\pi - \sigma}{2} \right| \tag{12}$$

In this way, the originally linear sharp corners (red) are replaced by gentle turns (green) - see Fig. 12.

ACKNOWLEDGMENT

The authors would like to acknowledge QUT Centre for Robotics for providing some of the funding support needed for this project.

REFERENCES

[1] D. Kirk, Optimal Control Theory: An Introduction.
[2] A. Aghajani, and A. Doustmohammadi, "Formation control of multi-vehicle systems using cooperative game theory," 2015
[3] M. Bloesch, S, Weiss, D. Scaramuzza and R. Siegwart, "Vision based MAV navigation in unknown and unstructured environments," 2010
[4] M. Burri, J. Nikolic, C. Hurzeler, G. Caprari and R. Siegwart, "Aerial service robots for visual inspection of thermal power plant boiler systems," *2nd International Conference on Applied Robotics for the Power Industry (CARPI)*, 2012
[5] A. Steinleitner, "Formation control for multi-rotor UAV systems," 2019
[6] F. Shamsi, "Time Varying Formation Control Using Differential Game Approach," 2011. – ISBN 9783902661937, S. 1114–1119
[7] W. Lin, Z. Qu and M. Simaan, "Distributed game strategy design with application to multi-agent formation control," 2014
[8] Z. Nikooeinejad, A. Delavarkhalafi and M. Heydari, "A numerical solution of open-loop Nash equilibrium in nonlinear differential games based on Chebyshev pseudo spectral method,"*Journal of Computational and Applied Mathematics*, vol. 300, pp. 369 - 384, 2016
[9] W. Lin, Differential Games For Multi-agent Systems Under Distributed Information.
[10] J. Marden, G. Arslan and J. Shamma, "Nonzero-sum differential games," *Journal of Optimization Theory and Applications*, vol. 3, pp. 184–206, 1969.
[11] D. Swaroop and J. Hedrick, "String stability of interconnected systems," *IEEE Transactions on Automatic Control*, vol. 41, No. 3, pp. 349–357, 1996
[12] X. Yang and G. Liu, "Output consensus conditions of linear MIMO multi-agent systems," *IEEE Access*, vol. 6, pp. 43672-43682, 2018