Journal of Intelligent and Robotic Systems, Volume 103, 2021, Article number 53
DOI:10.1007/s10846-021-01492-1

Using Lazy Agents to Improve the Flocking Efficiency of Multiple UAVs

Yeongho Song, Myeonggeun Gu, Joonwon Choi, Hyondong Oh*

School of Mechanical, Aerospace and Nuclear Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, Republic of Korea

Seunghan Lim

Pablo Air Co. Ltd., Republic of Korea

Hyo-Sang Shin, Antonios Tsourdos

Institute of Aerospace Sciences, Cranfield University, Cranfield MK43 0AL, UK

Abstract

A group of agents can form a flock using the augmented Cucker-Smale (C-S) model. The model autonomously aligns them to a common velocity and maintains a relative distance among the agents in a distributed manner by sharing the information among neighbors. This paper introduces the concept of inactiveness to the augmented C-S model for improving the flocking performance. It involves controlling the energy and convergence time required to form a stable flock. Inspired by the natural world where a few lazy (or inactive) workers are helpful to the group performance in social insect colonies. In this study, we analyzed different levels of inactiveness as a degree of control input effectiveness for multiple fixed-wing UAVs in the flocking algorithm. To find the appropriate inactiveness level for each flock member, the particle swarm optimization-based approach is used as the first step, based on the initial condition of the flock. However, as the significant computational burden may cause difficulties in implementing the optimization-based approach in real time, we also propose a

*Corresponding author
Email address: h.oh@unist.ac.kr (Hyondong Oh)

Published by Springer. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial License (CC:BY:NC 4.0).
The final published version (version of record) is available online at DOI:10.1007/s10846-021-01492-1. Please refer to any applicable publisher terms of use.

heuristic adaptive inactiveness approach, which changes the inactivity level of selected agents adaptively according to their position and heading relative to the flock center. The performance of the proposed approaches using the concept of lazy (or inactive) agents is verified with numerical simulations by comparing them with the conventional flocking algorithm in various scenarios.
Keywords: multi-agent system, flocking algorithm, augmented Cucker-Smale model, inactiveness, particle swarm optimization.

1. Introduction

Multi-agent systems have attracted considerable attention as they can improve the mission success rate, efficiency, and system autonomy. Cooperative or collective behavior of multi-agent systems can be achieved through interactions and consensus among agents in a distributed manner [21, 16]. Along with the increasing interest in autonomy, several researchers have tried to find efficient self-organization methods by observing the efficient system operation of natural organisms [2, 17, 9]. In unmanned aerial vehicle (UAV) operations, consensus-based cooperative behaviors represent a form of flocking, which is used in various tasks [38, 13]. Flocking (or loose formation) means that the UAVs satisfy the Reynolds flocking rules [27] where they mimic bird clustering in nature without a predetermined pattern in flight.

Using the Reynolds flocking rules, Vicsek showed that the random initial heading angle of the agents could be aligned in the same direction using a distributed method [31]; various studies have also been conducted on the Vicsek model [7, 26, 1]. To further develop the Vicsek model, Cucker and Smale designed a flocking model to reach the consensus of velocity [11], called the Cucker-Smale (C-S) model. Perea et al. [23] reported that applying the C-S model to spacecraft formation control has advantages such as decreased fuel consumption and maximum distance between spacecraft over the conventional control method. Shen et al. [28] showed that a hierarchical leader with a freewill acceleration has an advantage in terms of the convergence rate toward the flock-

ing state in the C-S model. Studies on the stability of the C-S model have been performed in an environment with disturbance [14, 12] or the constraints of flying at a constant speed [10]. Furthermore, as the safe operation between multiple UAVs is an important issue, in order to achieve collision avoidance in the flocking model, the singular interaction kernel was used in [5], and the augmented C-S model was proposed by adding a bonding force term to maintain the relative distance between neighbors in [22].

The original C-S type flocking model theoretically guarantees velocity consensus under specific conditions; however, forming a flock might require a large amount of energy and time caused by dynamic constraints or control saturation. This problem may occur rather frequently owing to the constraints of fixed-wing UAVs such as the maximum acceleration and turn rate and the restrictive convergence condition of the constant-speed C-S model [10]. To address these problems, we utilize inactiveness, which discounts the control input (as a degree of control effort effectiveness) generated by the existing flocking algorithm, yielding to improve performance. This approach is inspired by the habits of various species living together in nature. For instance, adopting a partially inactive state instead of the fully active state has been reported in ant colonies to increase labor efficiency and sustainability [24, 6, 15]. Wang et al. [32] applied this inactiveness concept to design a uniformly distributed circular formation of multiple particles in which decreasing the control input of a few particles improved the performance significantly.

We extend the above inactiveness concept to the flocking task of fixed-wing UAVs. To find the best inactiveness level for each flock member, social learning particle swarm optimization (SL-PSO) [8] is applied as the first step by using the initial condition of the flock and building upon our previous work [29, 18]. Although this optimization-based approach shows good performance, performing it in real time is difficult owing to significant computational burden. Besides, relying on a constant optimized inactivity level throughout the flocking operation may not provide sufficient robustness against uncertainty and disturbance. To overcome these limitations, we additionally propose a heuristic adaptive in-

activeness approach, which changes the inactivity level of the selected agents adaptively according to their position and heading relative to the flock center. Throughout this study, the effect of inactiveness was verified by the improvement in the energy as well as convergence time for the flocking task. This result, which is quite remarkable considering that the required convergence time and consumed energy for performing a certain control task generally have a trade-off relationship, promises great application potential in various research fields related to multiple agents.

This paper is structured as follows. In Sections II and III, we describe the flocking problem which demonstrates the effectiveness of the inactivity in a group. In Section IV, a technique for calculating the optimized inactivity level for each agent is presented, followed by a description of the adaptive inactiveness approach in Section V. In Section VI, we analyze the energy and convergence time efficiencies of the flocking task using the inactive group (termed as lazy group) and apply numerical simulations to compare the results with those achieved with a fully active group. Finally, conclusions and future work are presented in Section VII.

2. Problem Formulation

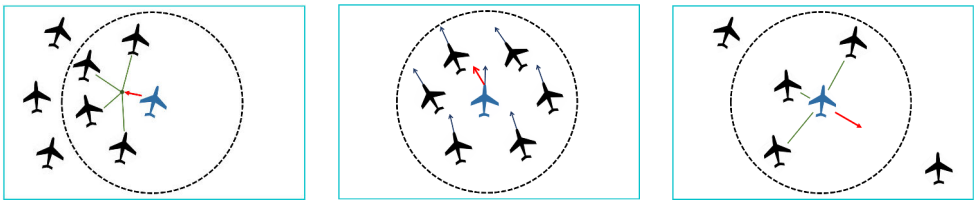
2.1. Overview of the Proposed Approach

In this study, we attempted to determine the effect of inactiveness in multi-agent systems while performing a flocking task in a 2-D space. The presence of lazy agents can exert changes on group behavior when compared with a fully active group (i.e., the conventional method) by reducing the control input proportional to the level of inactivity of an individual agent. The component of the inactivity level vector ($\mathbf{C}_{Lazy} = [C_{Lazy,1}, \cdots, C_{Lazy,N}]^T$) is calculated between zero and one for each agent where $\mathbf{C}_{Lazy}$ is the N dimensional vector and N is the number of agents. To evaluate the effectiveness of the flocking process, the cost function is defined according to the flocking performance in terms of control efforts and time required for the stable flocking convergence with

$\mathbf{C}_{Lazy}$. Based on the cost function, we first find the optimized $\mathbf{C}^*_{Lazy}$. Next, to overcome the limitation of the optimized approach, we propose a heuristic method in which $\mathbf{C}_{Lazy}$ changes adaptively. We will show that the group with lazy agents can achieve the flocking flight more efficiently compared with the fully active group for any initial positions $\mathbf{p}_0$ and velocities $\mathbf{v}_0$.

2.2. Flocking Task

For the safe and stable operation of a large number of agents, the flocking task is a fundamental element. To form a flock, agents should satisfy the Reynolds flocking rules [27], which comprise three elements: cohesion, alignment, and separation. Cohesion represents the group concentration. As depicted in Fig. 1(a), agents distant from the center of the group must move inward to the center to maintain a loose formation (i.e., flocking). Alignment depicted in Fig. 1(b) ensures that all agents maintain the same velocity. Even a few agents flying at a different velocity from others would cause a significant delay or even failure in flock formation. To prevent this, all agents in the group must align their velocity. Finally, separation depicted in Fig. 1(c) ensures collision avoidance among agents. If these three conditions are satisfied, a converged flocking state can be achieved from any arbitrary initial state, as shown in Fig. 2.



(a) Cohesion (b) Alignment (c) Separation

Figure 1: Reynolds flocking rules.

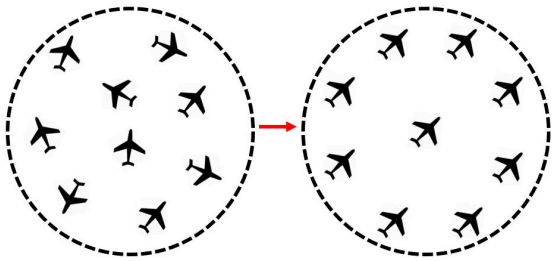


Figure 2: Achieving the flocking state from a random initial state.

3. Flocking Control Algorithm with Inactiveness

In this study, by focusing on a high-level flocking control design, a simple 2-D kinematics of the i-th agent moving at a constant speed is implemented as:

$$\begin{aligned} \dot{x}_i &= V\cos\theta_i, \\ \dot{y}_i &= V\sin\theta_i, \\ \dot{\theta}_i &= u_i, \quad \forall i = \{1, 2, \cdots, N\}. \end{aligned} \tag{1}$$

Here, x_i and y_i are the east and north displacements, respectively; V is the speed of the agent; θ_i is the heading angle; and u_i is the control input, which is constrained by the maximum turn rate (i.e., $|u_i| \leq u_{sat}$). Among the several flocking models developed for multi-agent systems, we adopted the C-S model [11]. The heading alignment of agents moving at a constant speed is calculated in the C-S model as [10]:

$$u_i^{CS} = \frac{\lambda}{N}\sum_{j=1}^{N}\psi(r_{ij})\sin(\theta_j - \theta_i), \tag{2}$$

where r_{ij} is the relative distance between the i-th and j-th agents, $\psi(r) = 1/(1+r^2)^{\beta}$, and $\lambda > 0$, $\beta \geq 0$ are constants. When $\beta < 1/2$ and the maximum difference in the initial heading angles of the agents is less than 90 degrees, alignment of their heading angles can be achieved. As the C-S model satisfies only the alignment condition among the Reynolds flocking rules, a term to maintain the relative distance among agents was added to ensure cohesion and

separation [22]. The additional term, referred to as the bonding force, acts according to the relative displacement of the agents, and the heading command can be derived as,

$$u_i^{Bonding} = \frac{\sigma}{NV}\sum_{j=1}^{N}\Big(\frac{K_1}{2r_{ij}^2}\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{p}_i - \mathbf{p}_j\rangle + \frac{K_2}{2r_{ij}}(r_{ij}-R)\Big)\langle [-\sin\theta_i, \cos\theta_i]^T, \mathbf{p}_j - \mathbf{p}_i\rangle, \tag{3}$$

where $\mathbf{p}_i = [x_i,\ y_i]^T$ and $\mathbf{v}_i = [\dot{x}_i,\ \dot{y}_i]^T$ are the position and velocity of the i-th agent, respectively; R is the parameter relating to the relative distance maintained; $\langle\cdot,\cdot\rangle$ is the inner product notation; and σ, K_1, and K_2 are positive constants. By combining the alignment term (2) and relative distance control term (3), the augmented C-S model with the saturation constraint can be given as,

$$u_i^{Flock} = u_i^{CS} + u_i^{Bonding}, \tag{4}$$

$$u_{i,sat}^{Flock} = \begin{cases} u_{sat}\mathrm{sgn}(u_i^{Flock}), & \text{if } \left|u_i^{Flock}\right| > u_{sat}, \\ u_i^{Flock}, & \text{otherwise.} \end{cases} \tag{5}$$

Here, $\mathrm{sgn}(\cdot)$ is the sign function.

The problem with the alignment of the heading angle is that the convergence conditions are difficult to fulfill under arbitrary initial conditions. Furthermore, even if the alignment condition is satisfied, desired flocking configuration is not guaranteed owing to the influence of saturation and the additional term for controlling the relative distance. These problems cause the agents to spend excessive energy and time in flocking. One way to address this problem is to apply the concept of inactiveness to the control input. As discussed in the Introduction, inactiveness is inspired by events observed in nature; that is, a few lazy insects improved the efficiency and sustainability of the whole group. By using the inactivity level, the i-th agent's heading angle is updated as,

$$\dot{\theta}_i = u_i = C_{Lazy,i}u_{i,sat}^{Flock}, \tag{6}$$

where $C_{Lazy,i} \in [0,1]$ is the i-th agent's inactivity level, and decreases its control input.

It is worthwhile noting that the collision avoidance among agents cannot be explicitly guaranteed by using the above flocking model in the transient period before converging to the final flock state depending on the initial configuration or communication topology. Although there are some recent approaches on collision avoidance for flocking using the potential field-based reactive control [18], distributed model predictive control [33] or reinforcement learning [35], in order to strictly (or explicitly) ensure collision avoidance in a complex and dynamic environments for flocking, more rigorous studies should be performed by adopting recent approaches [36, 37, 39]; this is beyond the scope of this paper since the main purpose of this study is to analyze the effect of the presence of lazy (i.e., inactive) agents on the flocking performance. Thus, this study assumes that the agents are separated by slightly different heights.

Besides, provided that the UAV has a low-level autopilot system, this study aims to design guidance command inputs for the flocking flight. With the time-scale separation principle [25, 3], assuming that the bandwidth of the low-level flight autopilot system is much faster (e.g. five to ten times) than that of the flocking guidance command, it is common to initially design and verify the guidance law and control algorithm separately. Therefore, like other literature [30, 20, 19] considering similar guidance problems, the simple dynamic model as in Eq. (1) with proper control saturation values could be used to design the flocking algorithm for fixed-wing UAVs. However, the final validation needs to be made with higher fidelity dynamic models and flight tests considering explicit collision avoidance among agents; these remain as future work.

4. Inactivity Optimization Using SL-PSO

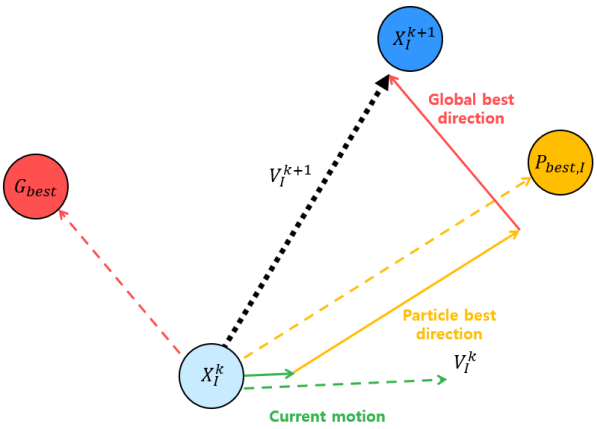
4.1. Optimization Method: SL-PSO

As the cost function to optimize $\mathbf{C}_{Lazy}$ is highly non-convex and difficult to solve analytically, we use the heuristic optimization algorithm which does not require gradient information. Among various heuristic optimization approaches, the particle swarm optimization (PSO) algorithm is known to provide

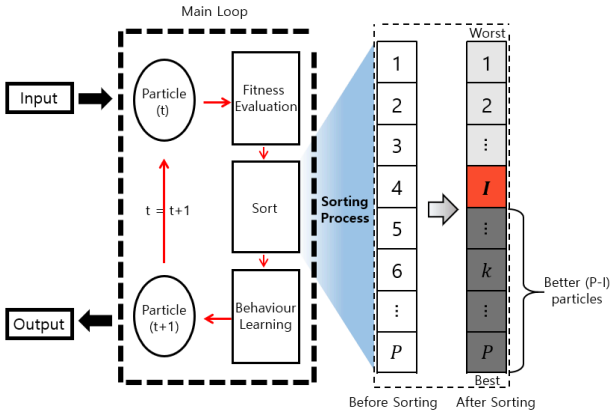
fast convergence close to the optimal solution. However, since the original PSO does not guarantee an optimal solution across the search domain, i.e., the solution obtained by the PSO is sub-optimal, we adopt the social learning particle swarm optimization (SL-PSO) to obtain the better solution among variations of the PSO [8]. The SL-PSO introduces sociological factors into the PSO where each particle probabilistically learns from one of better particles in the current swarm. Here, the concepts of the optimization method is briefly described.

Figure 3(a) illustrates the concept of PSO. The PSO algorithm is a computational method using candidate solution particles to optimize a problem in the search-space where the particles acquire the cost according to their position at every iteration. Each particle's movement is updated based on historical information, which is the direction of the weighted sum of the best solution of itself (called P_{best}), the global best solution by a whole swarm (G_{best}), and its current velocity.

The difference between PSO and SL-PSO is a learning process from other particles. Specifically, whole particles in the swarm learn from the global best solution at every iteration in PSO. However, whole particles are arranged based on their cost from the worst to the best, and each particle probabilistically learns from one of the better particles. In other words, the I-th ranked particle probabilistically learns from one of the $(P - I)$ particles that have a better cost as illustrated in Fig. 3(b). This can be interpreted as the addition of the sociological learning theory to the PSO method. Through this process, SL-PSO converges towards the optimal cost better [8]. Compared to the PSO algorithm, SL-PSO has a number of benefits. First, it has high computational efficiency and superior performance. Second, memory usage is low as there is no need for the past cost. Lastly, there is no burden of parameter setting which makes this algorithm can be easily used. The overall structure of SL-PSO is depicted in Fig. 3(b).



(a) PSO



(b) SL-PSO

Figure 3: Concept of the particle swarm optimization algorithm.

4.2. Cost Function

The cost function to evaluate the fitness is set to the measure of the flocking performance as,

$$J_{energy}(\mathbf{u}) = \frac{V}{N} \sum_{i=1}^{N} \int_0^{t_c} |u_i(\tau)| \, d\tau + \rho t_c, \tag{7}$$

where $\mathbf{u}$ is the control input vector of all agents, t_c is the convergence time, and ρ is a positive weighting parameter. Convergence is confirmed when the velocity deviation of the group narrows to a certain value close to zero and the position deviation of the agents is smaller than the predetermined threshold. The physical interpretation of $\mathbf{u}$ is the control effort required to change the heading angle, and ρt_c is the energy consumption for maintaining a constant speed that is assumed to be proportional to the time spent until convergence. Thus, J_{energy} represents the total energy consumption to reach a stable flocking state.

Accordingly, the optimization of the inactivity level can be expressed in the following form:

$$\mathbf{C}^*_{Lazy} = \arg \min_{\mathbf{C}_{Lazy} \in \mathbb{R}^N} J_{energy}(\mathbf{u}). \tag{8}$$

Here, $\mathbf{C}^*_{Lazy}$ is the N dimensional vector of $\mathbf{C}_{Lazy}$ with the best performance. To ensure that the performance of $\mathbf{C}^*_{Lazy}$ is better than that of the conventional method, one of the initial particles of SL-PSO is set to $\mathbf{C}^0_{Lazy} = \mathbf{1}$, which indicates the fully active state. Algorithm 1 shows the use of inactiveness for the optimization of flocking.

Algorithm 1: Optimization using SL-PSO

Set parameters ε, P (number of particles), and $\vartheta = \{1, 2, \cdots, P\}$
Set initial values $\mathbf{p}_0, \mathbf{v}_0$, $n = 0$, and $\mathbf{C}^I_{Lazy}(0), \forall I \in \vartheta$
while $\left\|\mathbf{C}^I_{Lazy}(n) - \mathbf{C}^J_{Lazy}(n)\right\| > \varepsilon, \forall I, J \in \vartheta$ **do**
    $J^I_{energy} = \text{Flocking}\big(\mathbf{p}_0, \mathbf{v}_0, \mathbf{C}^I_{Lazy}(n)\big), \forall I \in \vartheta$ by Eq. (7)
    $\mathbf{C}^I_{Lazy}(n+1)$ updates by the social learning process, $\forall I \in \vartheta$
    $n = n + 1$
end while
return $\mathbf{C}^*_{Lazy} = \mathbf{C}^P_{Lazy}(n)$

5. Heuristic Adaptive Inactiveness Approach

Although the optimization-based approach described in the previous section is expected to provide a promising performance, performing it in real time is difficult owing to a significant computational burden. Besides, the use of a constant optimized inactivity level $\mathbf{C}^*_{Lazy}$ based on the initial flock state during the entire flocking operation may not provide the expected flocking performance improvement in a real dynamic environment with uncertainty and disturbances. To overcome these limitations, in this section, we propose a heuristic method to determine the inactivity level $\mathbf{C}_{Lazy}$ adaptively according to the current flock configuration.

Notably, the previous optimization method determines the inactiveness for all the agents in the flock at different levels. However, looking at the ant colony in the natural world, only a certain portion of ants in the colony exhibit the so called "laziness" [6]. Motivated by this observation, we identified suitable agents to impose inactivity. Let us first visualize the general flocking task described in Section 2.2. In the early stages of the flocking process from the randomly distributed initial condition, all agents try to move towards the center of flock for cohesion. Once this is achieved to a certain extent, the agents start to focus on reaching velocity consensus (i.e., alignment). This can be observed in Fig. 4, which shows the time history of the control input (u_i^{Flock}) decomposed in terms

of cohesion ($u_i^{Bonding}$) and alignment (u_i^{CS}) during a few flocking simulations. As shown in this figure, in the early stage, most of the control efforts are dedicated toward controlling the relative distance (i.e., high $u_i^{Bonding}$). Thus, we selected agents initially expected to consume large amounts of energy for cohesion as the inactive agents. To this end, the fitness index $f_i(t)$ for the selection of inactive agents is calculated based on two indicators: the (normalized) relative distance $\alpha_i(t)$ of the current agent position to the center of the flock ($\mathbf{p}_c(t)$) and $\gamma_i(t)$, which corresponds to the angle $\varphi_i(t)$ between the current heading direction and the line connecting the current agent position and the flock center. This is shown in Fig. 5(a) and given by:

$$\alpha_i(t) = \begin{cases} \|\bar{\mathbf{r}}_i(0)\|, & \text{if } t = 0, \\ \frac{\|\bar{\mathbf{r}}_i(t)\|}{\max_{\tau\in[0,\ t]}\|\bar{\mathbf{r}}_i(\tau)\|}, & \text{if } t > 0, \end{cases} \tag{9}$$

$$\gamma_i(t) = \frac{1}{2}\big(1 - \cos\varphi_i(t)\big), \tag{10}$$

$$f_i(t) = \alpha_i(t)\gamma_i(t), \tag{11}$$

where $\bar{\mathbf{r}}_i(t) = \mathbf{p}_c(t) - \mathbf{p}_i(t)$. Notably, a high initial fitness index for an agent means that it is far away from the flock center, and its heading direction is quite different from that towards the flock center; therefore, it is expected to require high control efforts at the initial stage of the flocking process. Accordingly, an agent with a large $f_i(0)$ should be assigned a higher priority to impose the inactiveness, as illustrated in Fig. 5(b).

We now introduce a heuristic rule to change the inactivity level adaptively according to the flocking phase. As shown in Fig. 4, allowing a high inactivity level (i.e., low C_{Lazy} value) in the early flocking phase is effective for cohesion. However, in the alignment phase, which involves a fine tuning of the heading direction among the agents, a high inactivity level can degrade the velocity consensus performance. Hence, the inactivity level is adaptively determined using the fitness index as,

$$C_{Lazy,i}\,(t) = 1 - Gf_i(t),\ t > 0, \tag{12}$$

where $G \in (0, 1]$ is the gain that prevents $C_{Lazy,i}\,(t)$ from reducing to zero. It

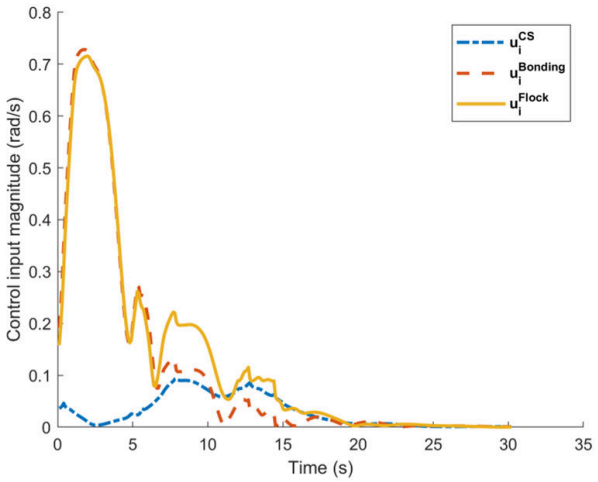
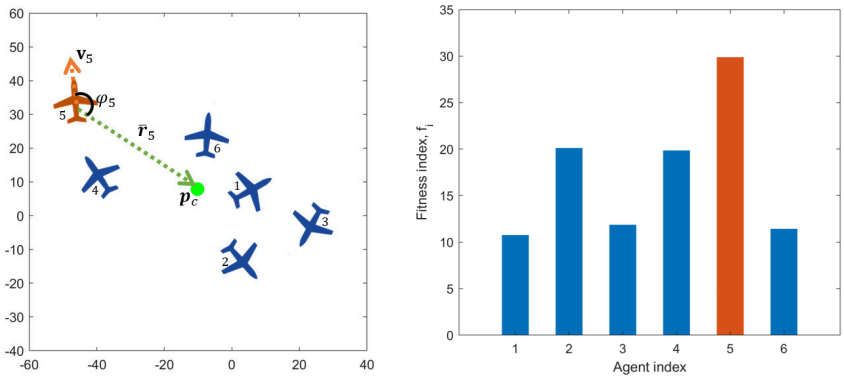


Figure 4: Time history of the control input magnitude during flocking simulation.



(a) Initial state of agents (b) Fitness index

Figure 5: Computation of fitness index.

should be noted that the adaptive inactivity level generally starts from a small value and increases as the flocking process progresses.

6. Numerical Simulation Results

In this section, simulation results for the augmented C-S model with inactivity are discussed. The initial conditions such as the heading angle and position of each agent were randomly set for 60 trials of Monte-Carlo simulation. The initial agent position is bounded within a square area whose edge length is L_{bound}. The parameter settings for the simulation are listed in Table 1. The movie clip for the simulations can be found at `https://www.youtube.com/watch?v=aOtsVTU_iOU`.

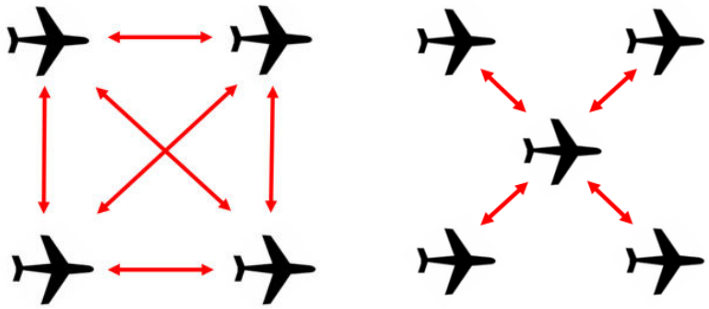
Table 1: Simulation parameters

Parameter	Value	Unit
Speed, V	15	m/s
Number of agents, N	20	
Predefined distance, R	60	m
Communication decay rate, β	1/3	
Energy coefficient to maintain a constant speed, ρ	1	
Inter agent strength, λ	5	
Bonding force strength, σ	1	
Constant gain, (K_1, K_2)	(1, 3)	
Inactiveness gain, G	0.8	
Maximum turn rate, u_{sat}	8/15	rad/s
Initial location bound, L_{bound}	250	m
Number of SL-PSO particle, P	100	

6.1. Flocking with Optimized Inactiveness

We consider two different scenarios based on the communication structures. In Fig. 6(a), the first scenario demonstrates the effect of inactiveness on each

agent, with a fully connected and undirected network topology, in which the information of all agents in the group is shared. On the other hand, the second scenario considers a central communication network topology, as shown in Fig. 6(b). In this case, each agent communicates through the central agent. This scenario exhibits the highest degree of centrality in the network, whereas other factors are similar to the first scenario. This centrality communication structure could play an important role in a hierarchical or leader-following multi-agent system.



(a) Fully connected network (b) Centrality network

Figure 6: Communication structures of the two scenarios.

6.1.1. Fully Connected Network Case

In this subsection, the simulation results of the first scenario are discussed. We analyzed the components of $\mathbf{C}^*_{Lazy}$ sorted in ascending order to focus on the distribution tendency of inactivity levels for achieving efficient flocking as shown in Fig. 7. Except for a few outliers, the average value of $\mathbf{C}^*_{Lazy}$ tends to lie between 0.4 and 1, which indicates that each agent in the group should have a proper inactivity level to improve the group performance instead of random inactiveness.

Table 2 demonstrates the performance improvement in the optimized lazy group compared with that in the fully active group (with $\mathbf{C}_{Lazy} = \mathbf{1}$). Contrary to the expectation that the agent inactivity caused by $\mathbf{C}_{Lazy}$ would reduce the

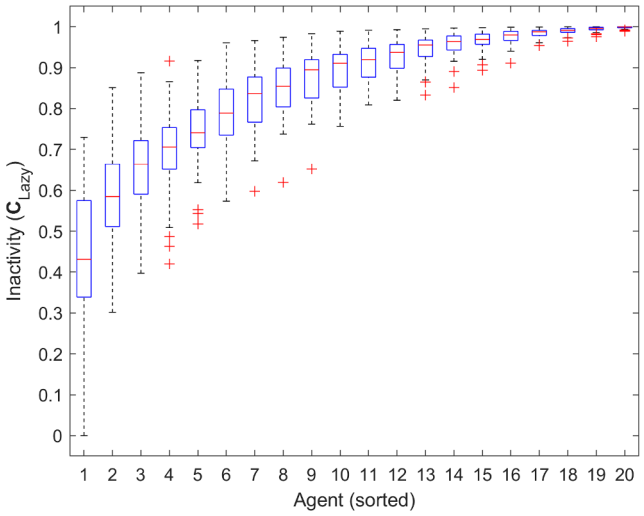


Figure 7: Inactivity levels (sorted in the ascending order) in the fully connected network.

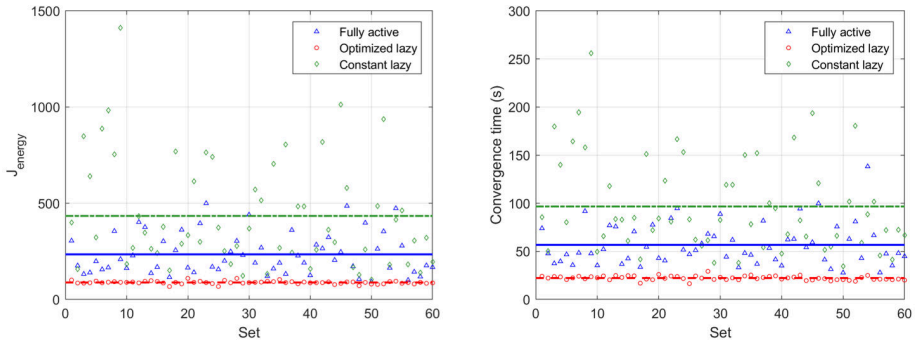
flocking efficiency, the result in Table 2 shows that both energy consumption and convergence time reduces remarkably when the optimized inactivity is applied.

Table 2: Performance in the fully connected network

	Mean J_{energy}	**Mean convergence time**
Fully active group	234.5	56.7
Optimized lazy group	88.2	22.3
Improvement ratio	**62.4%**	**60.7%**

In Fig. 8, the flocking performance is analyzed in terms of consumed energy and flocking convergence. In the figures, the blue line indicates the average performance of the fully active groups whereas the red dotted line is for optimized lazy groups. The result shows the benefit of $\mathbf{C}_{Lazy}$ for any type of initial en-

vironment. To compare the performance of the optimized $\mathbf{C}^*_{Lazy}$ (different for each agent) with the average of $\mathbf{C}^*_{Lazy}$ (fixed for all agents), the average $\mathbf{C}^*_{Lazy}$ is set to C^{Avg}_{Lazy} as 0.8 and applied to all agents in the flocking simulation. This result is denoted as a green line and shows that the performance is much worse than that of the fully active group, which implies that a group of agents with the same level of inactivity does not produce any benefit. This result contrasts with the characteristics of inactivity observed in the natural world; the effect is similar to just adjusting (lowering) the control gain.

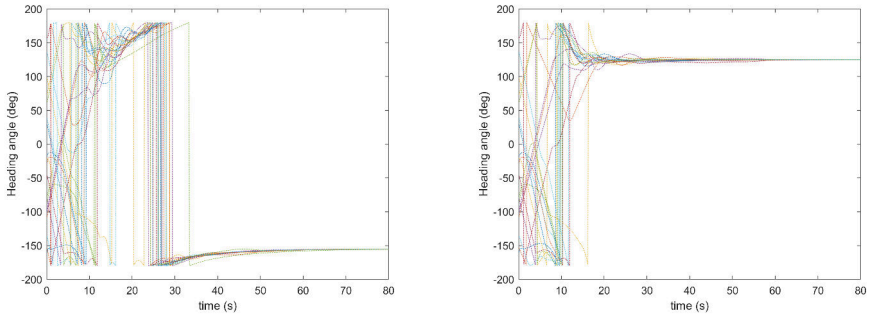


(a) Energy (b) Convergence time

Figure 8: Comparison of flocking performance in the fully connected networks.

Moreover, the effect induced by the inactive agents can also be observed from the sample heading angle changes, as shown in Fig. 9. In the case of the fully active group in Fig. 9(a), each agent actively changes its state according to that of its neighbor. However, if the control input exceeds the saturation limit, the agents will not reach the desired state quickly, causing unnecessary energy consumption. Besides, agents reacting too sensitively to their neighbors frequently disturb the flocking, increasing the required time for convergence. On the other hand, agents respond less sensitively owing to their inactivity, as shown in Fig. 9(b). As these agents are less sensitive to the state information of their neighboring agents, they tend to follow already formed clusters with little fluctuation.

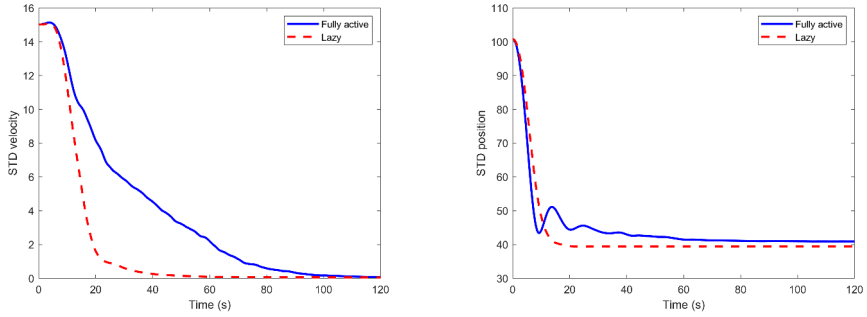
Figure 10 shows the average standard deviations of the velocity and position,



(a) Fully active group (b) Optimized lazy group

Figure 9: Time history of the heading angles in the fully connected network.

which indicate the quality of the flock formation. The velocity and position deviations correspond to the alignment and the cohesion and separation of the Reynolds flocking rules, respectively. In Fig. 10(a), the red dotted line of the optimized lazy group decreases with time much faster than the blue line of the fully active group. The reason for the high standard deviation of the velocity at the initial stage is that the agents move toward the center of the group to get the desired distance among themselves. For the position deviation, the optimized lazy group shows better results without the fluctuation observed in the fully active group.



(a) Standard deviation of the velocity (b) Standard deviation of the position

Figure 10: Comparison of standard deviations in the fully connected network.

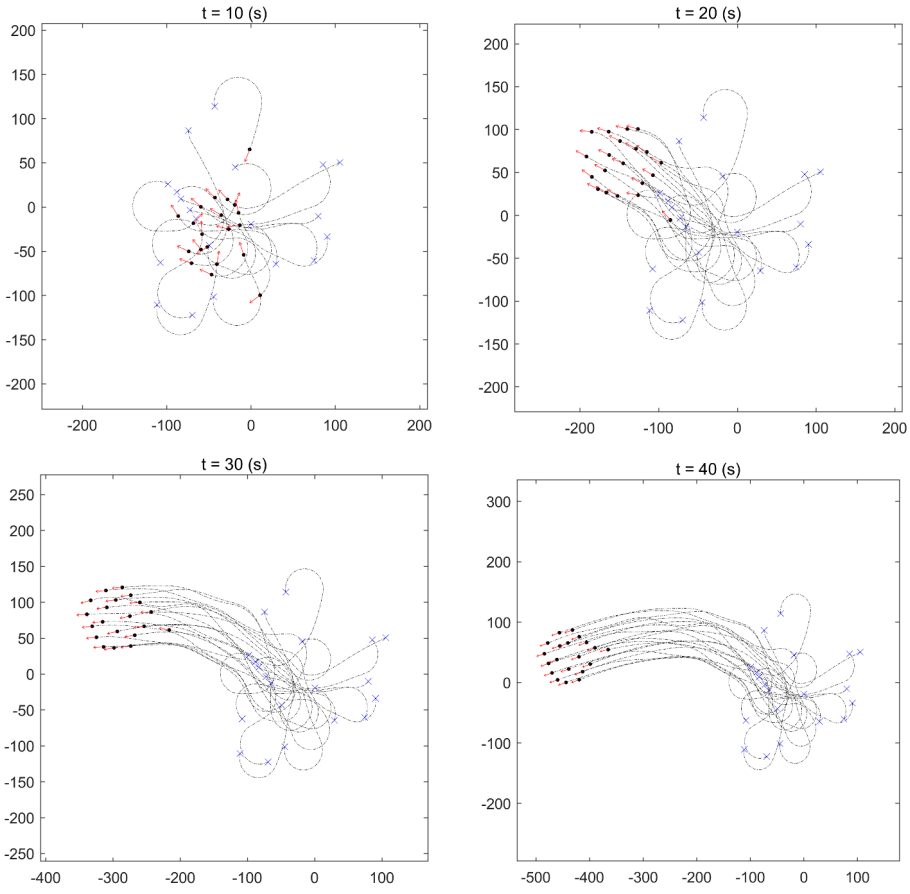


Figure 11: Trajectory of the fully active group in the fully connected network.

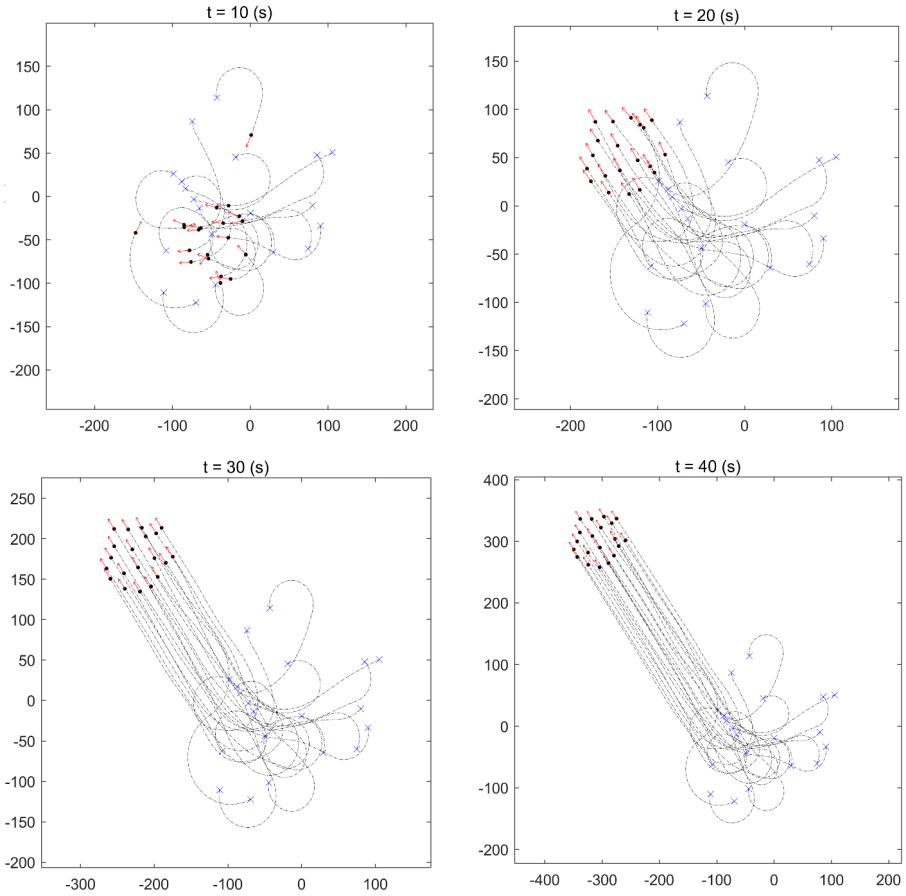


Figure 12: Trajectory of the optimized lazy group in the fully connected network.

Figures 11 and 12 show the sample trajectories of the agents over time. Through a trajectory of 10 s, the agents tend to move to the center of the group with a certain turn radius limit. The flocking of the fully active group takes approximately 30 s, while the optimized lazy group takes approximately 20 s. In the convergence state, both groups show a loose flocking pattern and exhibit a lattice formation. Notably, this formation pattern is affected by the bonding force that maintains the distance between the agents.

6.1.2. Centrality Network Case

In this scenario, only the agent with the highest degree of centrality communicates and exchanges the state information with the other members in the group, as shown in Fig. 6(b). In the simulation, the inactivity levels in the simulation result were sorted in ascending order to check the distribution tendency. As shown in Fig. 13, the central agent, marked as agent number 1, has a significantly lower average inactivity level than all others. This phenomenon implies that the central agent tends to have a low inactivity level regardless of the initial state to achieve a better flocking performance.

In Table 3, the performance comparison results of the second scenario simulation are summarized. Similar to the previous scenario, the optimized lazy group significantly reduces the consumed energy and convergence time for the flocking convergence. Figure 14 shows the numerical results for each set. The blue and red lines indicate the average performance of the fully active and optimized lazy groups, respectively. The green dots represent the performance when the central and the other agents are assigned constant inactivity levels of 0.14 and 0.66 as the average $\mathbf{C}^*_{Lazy}$ values for the central agent and all the other agents, respectively. In contrast with the previous result where all the agents had the same constant inactivity level, (Fig. 8), this result shows that the group with constant inactivity levels has advantages in a central network in terms of the energy consumption and convergence time when compared with the fully active groups even without any optimization. Because the inactive central agent is less affected by the neighbors, it performs the role of a convergence

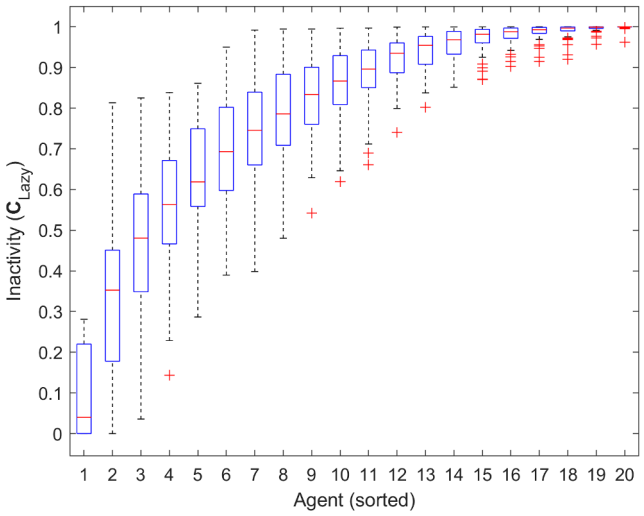


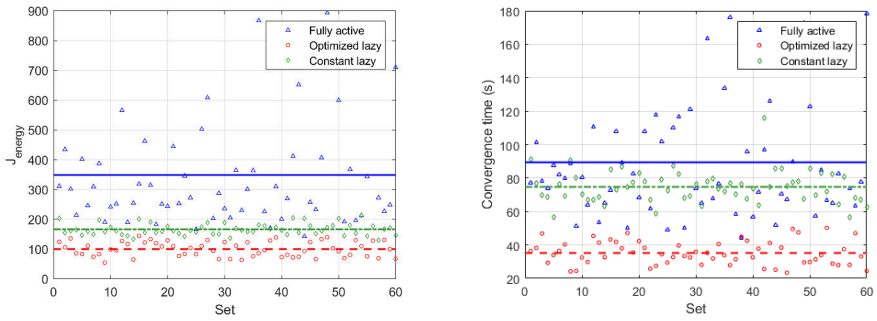
Figure 13: Inactivity level (sorted in ascending order except for the central (1st) agent) in the centrality network.

point which helps to achieve a more efficient consensus.

Table 3: Performance in the centrality network

	Mean J_{energy}	**Mean convergence time**
Fully active group	349.1	89.4
Optimized lazy group	100.3	35.2
Improvement ratio	**71.3%**	**60.6%**

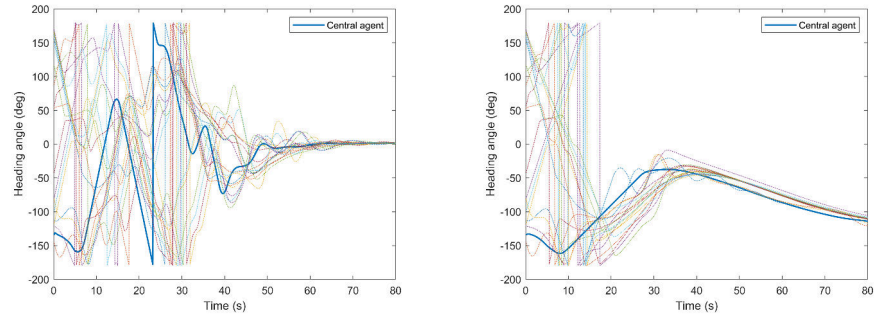
Figure 15 shows the changes in the sample heading angle with time. The thick blue line in Fig. 15(a) indicates the heading angle of the central agent, which changes significantly owing to the influence of neighboring agents. This change leads to a successive change in the heading angle of the other agents as well and requires substantial energy to perform a flocking task. In the case of the



(a) Energy (b) Convergence time

Figure 14: Comparison of flocking performance in the centrality network.

lazy group, however, the central agent tends to be less sensitive to the states of the neighboring agents, and as it maintains the low control input without rapid changes; eventually the other agents can easily follow the central agent.

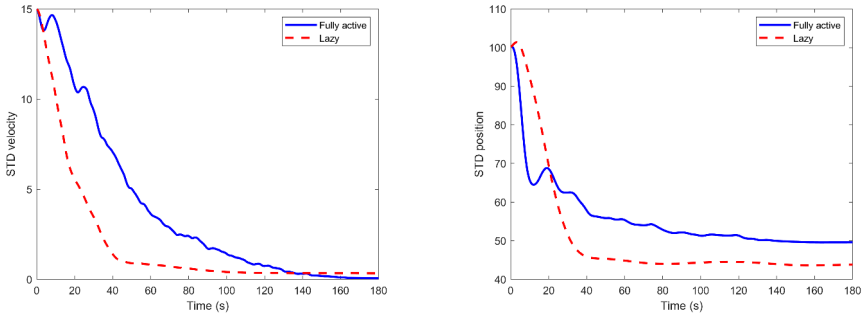


(a) Fully active group (b) Optimized lazy group

Figure 15: Time history of heading angles in the centrality network.

Figure 16 shows the velocity and the position deviations with time. Although the deviations show a continuously decreasing trend, the blue lines for the fully active group show a certain degree of fluctuation in both cases. On the other hand, the red dotted lines for the lazy group show a tendency to decrease smoothly and quickly without fluctuation. Figures 17 and 18 show the sample trajectories of the fully active and lazy cases, respectively, with a green line

depicting the central agent. The positions of the converged agents indicate that the central agent plays a critical role in both the fully active and lazy groups.



(a) Standard deviation of the velocity (b) Standard deviation of the position

Figure 16: Comparison of standard deviations in the the centrality network.

In the flocking task under the centrality network topology, it is efficient for the central agent to be inactive. This result suggests that an inactive central agent can benefit the overall performance when the information is concentrated in one node on a platform with a distributed control system.

6.2. Flocking with Adaptive Inactiveness

This subsection presents numerical simulation results by applying the heuristic adaptive inactiveness approach. Notably, we only considered the fully connected network case because the proposed approach was developed without considering the existence of a central agent (present in the centrality network). First, to determine the appropriate number of inactive agents in the flock, the performance improvement ratio is analyzed depending on the ratio of inactive agents to the entire group (20, 30, or 40 agents), as shown in Fig. 19. For this analysis, inactive agents are selected based on their fitness index $f_i(0)$, calculated with Eq. (11) because the agent with a high initial fitness index is expected to reduce the control efforts significantly by becoming inactive. For instance, the simulation result for 20 percent inactive agents among 20 agents in Fig. 19 is obtained by selecting 4 agents with the highest fitness index values. Once

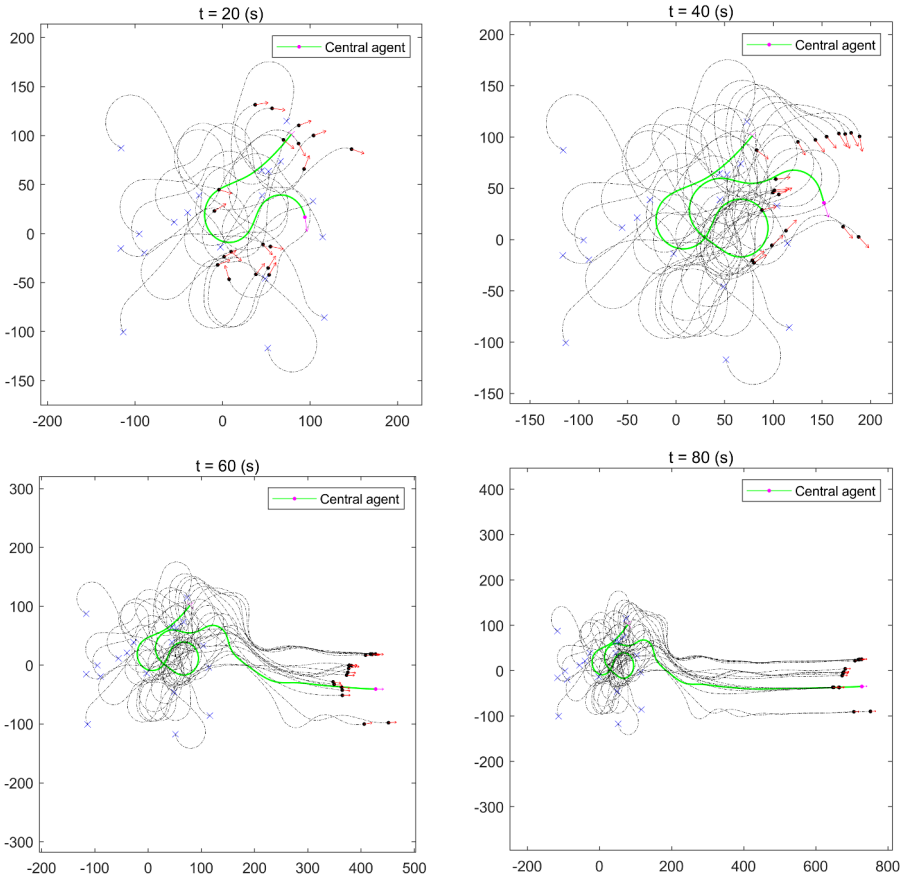


Figure 17: Trajectory of the fully active group in the centrality network.

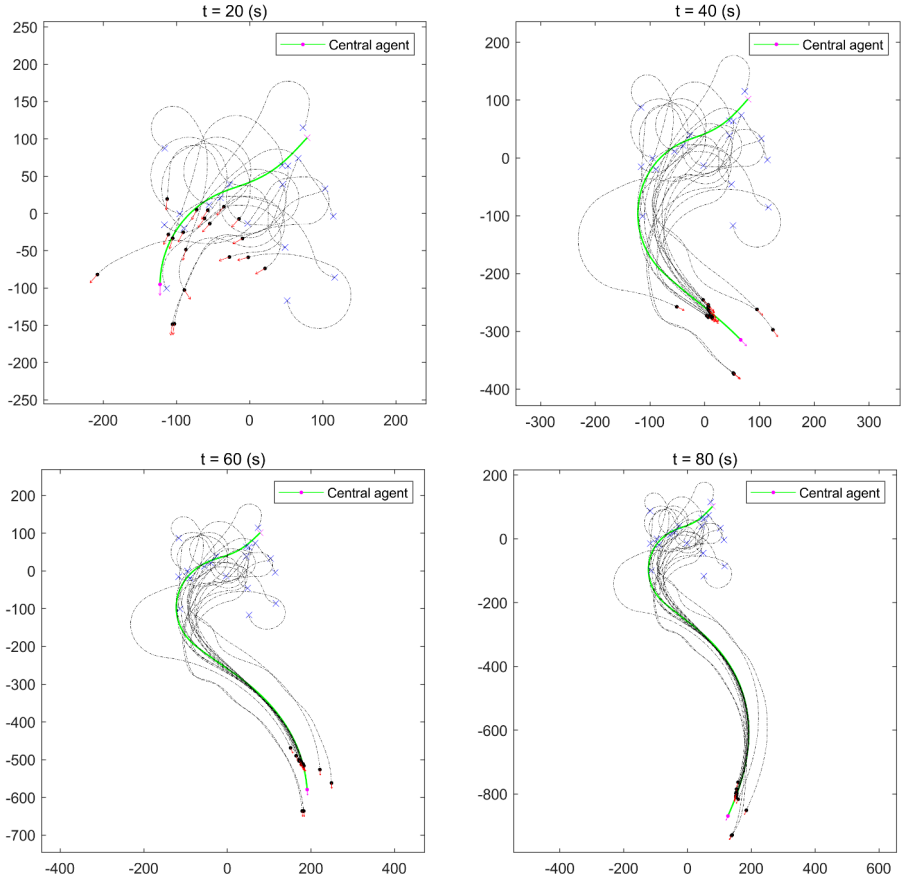


Figure 18: Trajectory of the optimized lazy group in the centrality network.

selected as inactive agents, they follow the adaptive inactiveness rule given in Eq. (12). Figure 19 shows that the use of a single highly inactive agent for the group of 20 agents results in the best performance, and the performance gradually decreases with an increasing number of inactive agents. This result may seem surprising at first; however, we should realize that more than one highly inactive agents distant from the flock center will delay the cohesion process by initially not joining the group common behavior; this delay starts and produces adverse effects that cannot be overcome by saving the initial control efforts for cohesion with inactiveness. Following the above analysis, we conducted simulations by selecting one agent out of 20 agents with the highest $f_i(0)$ as the inactive agent and applying the corresponding $C_{Lazy,i}$ value in Eq. (12).

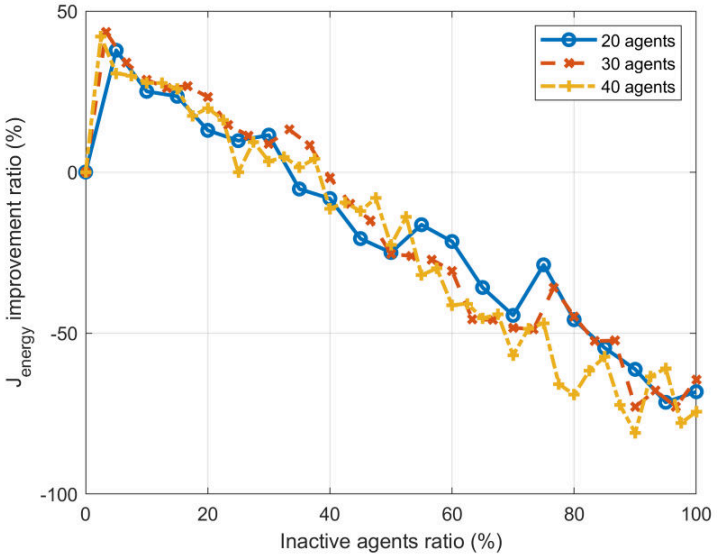


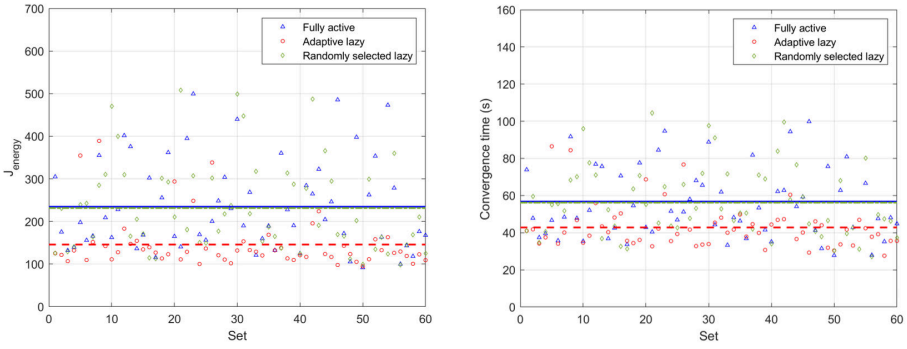
Figure 19: Performance improvement ratio with varying number of inactive agents.

Table 4 and Fig. 20 demonstrate the superior performance of the adaptive lazy group (with one inactive agent) compared with that of the fully active group. In the figures, the blue and red dotted lines indicate the averaged performance of the fully active and adaptive lazy groups, respectively. To verify the validity of the proposed approach that involves selecting the inactive agent

with the highest $f_i(0)$, the simulation was performed with one randomly selected inactive agent as well. The result is indicated with a green dotted line in the figure and is almost similar to that of the fully active group. This has two important implications: (i) selecting the inactive agent according to the initial fitness index (expected to consume excessive energy for cohesion) is validated; and (ii) the risk of divergence (i.e., failure in the flocking state) is low in the proposed inactivity approach because we obtain results similar to those in the fully active group even with a randomly-selected inactive agent.

Table 4: Performance in the adaptive method

	Mean J_{energy}	**Mean convergence time**
Fully active group	234.5	56.7
Adaptive lazy group	145.6	42.8
Improvement ratio	**37.9%**	**24.5%**

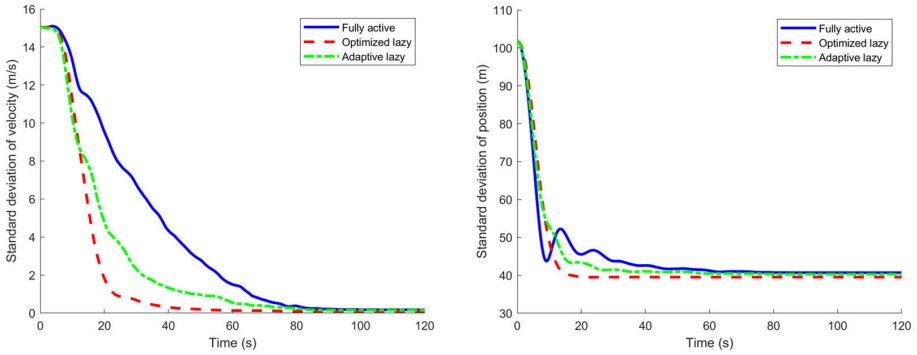


(a) Energy (b) Convergence time

Figure 20: Comparison of flocking performance in the adaptive inactiveness approach.

Figure 21 shows the averaged standard deviations in velocity and position. The red dotted line of the optimized lazy group reaches zero most quickly, followed by that of the adaptive lazy group and the fully active group. Notably,

the result of the optimized lazy group is also included here for comparison purpose. Besides, Fig. 22 shows the sample trajectories of the adaptive lazy group with time. The green line for the selected inactive agent with the highest $f_i(0)$ shows distinctive behavior compared with the other fully active agents. Based on the adaptive inactiveness strategy, the inactive agent does not actively align its heading direction with the others during the early stages of the flocking process with a low C_{Lazy} value. Once cohesion is achieved to a certain degree, the inactive agent joins the group behavior with a high C_{Lazy} value, implying that it becomes almost fully active.



(a) Standard deviation of the velocity (b) Standard deviation of the position

Figure 21: Comparison of standard deviations in the adaptive inactiveness approach.

Note that the preceding simulations assumed the ideal communication situation, however, various communication problems may occur in real applications. To verify the effectiveness of the proposed algorithm in poor communication environments, the flocking performance of the fully active group and the adaptive lazy group are compared considering packet loss. Here, packet loss is defined as probabilistic communication failure and when a communication failure occurs, the agent is assumed to maintain the previous heading direction. Figure 23 shows that both groups have performance degradation as the percentage of the packet loss increases. However, the lazy group's performance is much better than that of the fully active group; since the lazy agents responds less sensi-

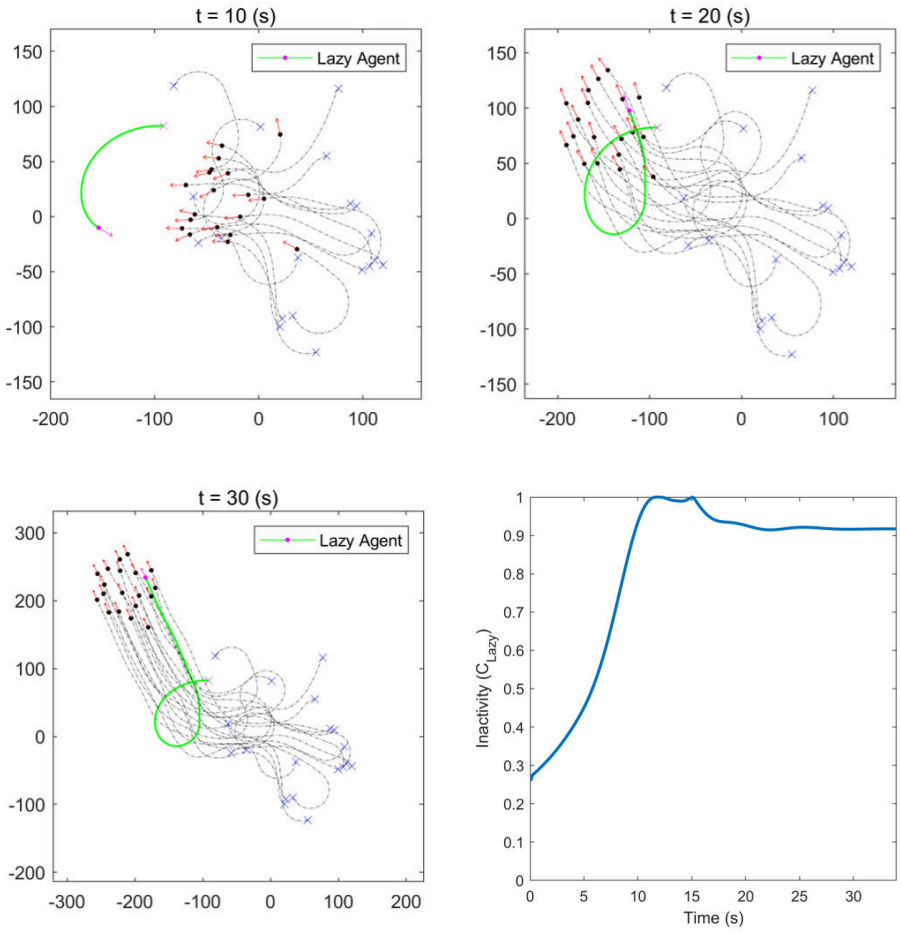


Figure 22: Trajectory of the adaptive lazy group with the inactivity level.

tively to their neighbors, intermittent communication failure has less impact on the lazy group. From this result, the proposed algorithm could be considered as more robust in a real environment where the communication condition is not ideal.

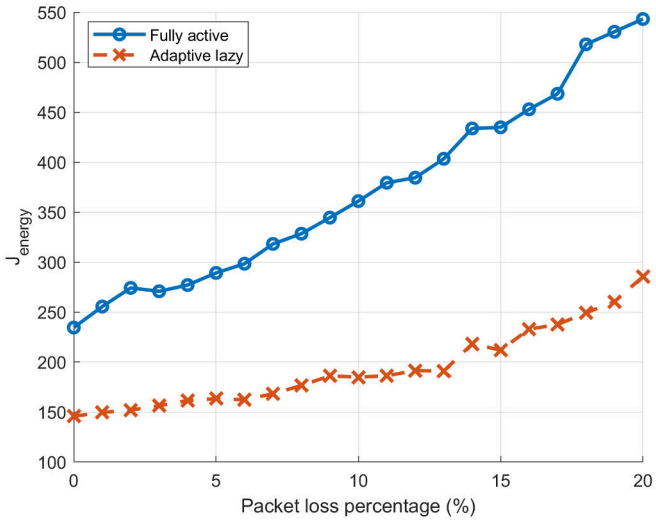


Figure 23: Performance comparison for packet loss.

7. Conclusions and Future Work

In this study, we demonstrated that the inactiveness of a few agents in a group can improve the efficiency of the flocking tasks of fixed-wing unmanned aerial vehicles by using the constant speed version of the augmented Cucker-Smale model. By applying social learning particle swarm optimization, optimized inactivity level tendency for ensuring the best flocking performance with respect to energy consumption and convergence time was confirmed. Then, we proposed a heuristic adaptive inactiveness method that selects appropriate inactive agents and changes the inactiveness level adaptively according to the flock configuration.

As future work, we will perform more rigorous theoretical analysis of the

flocking performance and convergence using the inactiveness concept proposed in this paper by adopting the advantages of the hierarchical leader [28] and pinning control [34] in which a few agents (e.g., informed agents) use different control inputs compared with the rest of the agents in the group; this concept is similar to the proposed inactiveness. Besides, we will study how to determine the level of inactivity more systematically by taking into account not only the initial/current position and velocity but also the network centrality (e.g., degree, closeness, and betweenness centrality [4]), which measures the importance of the node for propagating information in the network.

Ethical Approval

Not applicable

Consent to Participate

Not applicable

Consent to Publish

Not applicable

Authors Contributions

- Yeongho Song - Conceptualization, Methodology, Investigation, Original Draft Preparation
- Myeonggeun Gu Conceptualization, Methodology, Investigation, Original Draft Preparation
- Joonwon Choi Conceptualization, Methodology, Investigation
- Hyondong Oh: Conceptualization, Methodology, Supervision, Review & Editing

- Seunghan Lim: Conceptualization, Review & Editing
- Hyo-Sang Shin: Conceptualization, Review & Editing
- Antonios Tsourdos: Conceptualization, Review & Editing

Funding

This research has been supported by the Defense Challengeable Future Technology Program of Agency for Defense Development, Republic of Korea and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2020R1A6A1A03040570).

Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and materials

None

Bibliography

[1] Baglietto, G., and Albano, E. V. Nature of the order-disorder transition in the vicsek model for the collective motion of self-propelled particles. *Physical Review E 80*, 5 (2009), 050103.

[2] Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., and Procaccini, A. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the National Academy of Sciences 105*, 4 (2008), 1232–1237.

[3] Beard, R., and McLain, T. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.

[4] Borgatti, S. P. Centrality and network flow. *Social Networks 27*, 1 (2005), 55–71.

[5] Carrillo, J. A., Choi, Y.-P., Mucha, P. B., and Peszek, J. Sharp conditions to avoid collisions in singular cucker–smale interactions. *Nonlinear Analysis: Real World Applications 37* (2017), 317–328.

[6] Charbonneau, D., and Dornhaus, A. Workers 'specialized'on inactivity: behavioral consistency of inactive workers and their role in task allocation. *Behavioral Ecology and Sociobiology 69*, 9 (2015), 1459–1472.

[7] Chaté, H., Ginelli, F., Grégoire, G., Peruani, F., and Raynaud, F. Modeling collective motion: variations on the vicsek model. *The European Physical Journal B 64*, 3-4 (2008), 451–456.

[8] Cheng, R., and Jin, Y. A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences 291* (2015), 43–60.

[9] Choi, J., Oh, S., and Horowitz, R. Distributed learning and cooperative control for multi-agent systems. *Automatica 45*, 12 (2009), 2802–2814.

[10] Choi, S.-H., and Ha, S.-Y. Emergence of flocking for a multi-agent system moving with constant speed. *Communications in Mathematical Sciences 14*, 4 (2016), 953–972.

[11] Cucker, F., and Smale, S. Emergent behavior in flocks. *IEEE Transactions on Automatic Control 52*, 5 (2007), 852–862.

[12] Erban, R., Haskovec, J., and Sun, Y. A cucker–smale model with noise and delay. *SIAM Journal on Applied Mathematics 76*, 4 (2016), 1535–1557.

[13] Fu, X., Pan, J., Wang, H., and Gao, X. A formation maintenance and reconstruction method of uav swarm based on distributed control. *Aerospace Science and Technology* (2020), 105981.

[14] Ha, S.-Y., Lee, K., and Levy, D. Emergence of time-asymptotic flocking in a stochastic cucker-smale system. *Communications in Mathematical Sciences 7*, 2 (2009), 453–469.

[15] Hasegawa, E., Ishii, Y., Tada, K., Kobayashi, K., and Yoshimura, J. Lazy workers are necessary for long-term sustainability in insect societies. *Scientific Reports 6* (2016), 20846.

[16] Lang, X., and de Ruiter, A. A control allocation scheme for spacecraft attitude stabilization based on distributed average consensus. *Aerospace Science and Technology* (2020), 106173.

[17] Leitão, P., Barbosa, J., and Trentesaux, D. Bio-inspired multi-agent systems for reconfigurable manufacturing systems. *Engineering Applications of Artificial Intelligence 25*, 5 (2012), 934–944.

[18] Lim, S., Song, Y., Choi, J., Myung, H., Lim, H., and Oh, H. Decentralized hybrid flocking guidance for a swarm of small uavs. In *Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)* (2019).

[19] Oh, H., Kim, S., Shin, H.-s., and Tsourdos, A. Coordinated standoff tracking of moving target groups using multiple uavs. *IEEE Transactions on Aerospace and Electronic Systems 51*, 2 (2015), 1501–1514.

[20] Oh, H., Shin, H.-S., Kim, S., and Chen, W.-H. Communication-aware trajectory planning for unmanned aerial vehicles in urban environments. *Journal of Guidance, Control, and Dynamics 41*, 10 (2018), 2271–2282.

[21] Olfati-Saber, R., Fax, J. A., and Murray, R. M. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE 95*, 1 (2007), 215–233.

[22] Park, J., Kim, H. J., and Ha, S.-Y. Cucker-smale flocking with inter-particle bonding forces. *IEEE Transactions on Automatic Control 55*, 11 (2010), 2617–2623.

[23] Perea, L., Elosegui, P., and Gómez, G. Extension of the cucker-smale control law to space flight formations. *Journal of Guidance, Control, and Dynamics 32*, 2 (2009), 527–537.

[24] Pinter-Wollman, N., Hubler, J., Holley, J.-A., Franks, N. R., and Dornhaus, A. How is activity distributed among and within tasks in temnothorax ants? *Behavioral Ecology and Sociobiology 66*, 10 (2012), 1407–1420.

[25] Reiner, J., Balas, G. J., and Garrard, W. L. Flight control design using robust dynamic inversion and time-scale separation. *Automatica 32*, 11 (1996), 1493–1504.

[26] Ren, W., and Beard, R. W. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control 50*, 5 (2005), 655–661.

[27] Reynolds, C. W. Flocks, herds and schools: A distributed behavioral model. *Computer Graphics (ACM SIGGRAPH '87 Conf. Proc.) 21*, 4 (1987), 25–34.

[28] Shen, J. Cucker–smale flocking under hierarchical leadership. *SIAM Journal on Applied Mathematics 68*, 3 (2007), 694–719.

[29] Song, Y., Choi, J., Oh, H., Lee, M., Lim, S., and Lee, J. Improvement of decentralized flocking flight efficiency of fixed-wing uavs using inactive agents. In *AIAA Scitech Forum* (2019).

[30] Sun, D., Kwon, C., and Hwang, I. Hybrid flocking control algorithm for fixed-wing aircraft. *Journal of Guidance, Control, and Dynamics 42*, 11 (2019), 2443–2455.

[31] Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters 75*, 6 (1995), 1226–1229.

[32] Wang, C., and Xie, G. Lazy workers benefit group performance in circle formation tasks. *IFAC-PapersOnLine 50*, 1 (2017), 10383–10388.

[33] Wang, Y., Wang, X., Zhao, S., and Shen, L. A hierarchical collision avoidance architecture for multiple fixed-wing uavs in an integrated airspace. In *21st IFAC World Congress* (2020). arXiv:2005.14455.

[34] Xing, W., Shi, P., Agarwal, R. K., and Zhao, Y. A survey on global pinning synchronization of complex networks. *Journal of the Franklin Institute 356*, 6 (2019), 3590–3611.

[35] Yan, C., Xiang, X., Wang, C., and Lan, Z. Collision-free flocking with a dynamic squad of fixed-wing uavs using deep reinforcement learning. arXiv:2101.08074.

[36] Yu, X., and Zhang, Y. Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. *Progress in Aerospace Sciences 74* (2015), 152–166.

[37] Yu, X., Zhou, X., and Zhang, Y. Collision-free trajectory generation and tracking for uavs using markov decision process in a cluttered environment. *Journal of Intelligent & Robotic Systems 93*, 1 (2019), 17–32.

[38] Zhen, Z., Yan, C., Liangdong, W., and Bing, H. An intelligent cooperative mission planning scheme of uav swarm in uncertain dynamic environment. *Aerospace Science and Technology* (2020), 105826.

[39] Zhou, X., Yu, X., and Peng, X. UAV collision avoidance based on varying cells strategy. *IEEE Transactions on Aerospace and Electronic Systems 55*, 4 (2018), 1743–1755.



https://dspace.org/

Using lazy agents to improve the flocking efficiency of multiple UAVs

Song, Yeongho; Gu, Myeonggeun; Choi, Joonwon; Oh, Hyondong; Lim, Seunghan ...

2021-10-27
Springer
https://doi.org/10.1007/s10846-021-01492-1
Song Y, Gu M, Choi J, et al., (2021) Using lazy agents to improve the flocking efficiency of multiple UAVs. Journal of Intelligent and Robotic Systems, Volume 103, Issue 3, November 2021, Article number 53

Downloaded from a DSpace repository. https://dspace.org/