

## Decentralised aerial swarm for adaptive and energy efficient transport of unknown loads

Kangyao Huang, Jingyu Chen, John Oyekan\*

Department of Automatic Control and Systems Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK



### ARTICLE INFO

#### Keywords:

Aerial swarm robotics  
Cooperative transport  
Decentralized system  
Flocking  
Bacterium chemotaxis behavior

### ABSTRACT

Cooperative transport by a swarm of Quadcopters offers more flexibility and performance when carrying loads that are complex in structural profile and mass. Ensuring that team members of the swarm are optimally placed on these loads as well as able to resist disturbances from the environment during transport are current research challenges. In this paper, we present a decentralized behaviour based subsumption architecture for enabling a swarm of Quadcopters to explore an unfamiliar area, find a load and transport it to a target location cooperatively. In the architecture, three behaviours were used: an obstacle avoidance behaviour to avoid collisions with objects in the environment, a flocking behaviour to ensure swarm structure and a bacterium behaviour for exploration of the environment and to adapt to the mass profile of various detected loads.

By adapting to the mass profile of a detected load, we show that our architecture ensures even energy distribution among Quadcopters while achieving robustness to disturbances from the environment. Our results show that a mass adapting swarm is able to conserve energy during payload transportation when compared to a swarm that does not adapt to a load's profile. Furthermore, we do not use explicit communication between team members but instead rely on data from visual sensors attached to the Quadcopters. We experiment with simulations in a physics informed robot simulator called CoppeliaSim and demonstrate the effectiveness of our architecture when utilized for cooperative transport of irregular loads.

### 1. Introduction

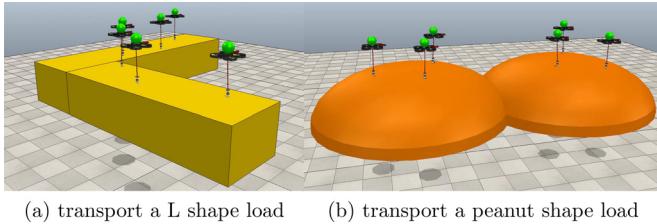
Rotary aerial robotic systems can provide increased efficiency and productivity in many sectors [1]. Due to their excellent control performance and convenience of vertical take-off and landing, rotary aerial robotic systems have gained widespread adoption in various sectors such as cargo carrying, package delivery and warehousing management. In order to cope with increasingly complex and varied load transport scenarios, the capability of a single robot can no longer meet many requirements (Fig. 1). Thus, the current consensus is that multiple agent systems can offer an increased performance on flexibility, robustness and payload weight when compared with one single capable robot [2,3]. Through cooperation of multiple individuals, larger loads with complex shapes can be transported. Nevertheless, the coordination and control of multiple aerial vehicles raises challenges that have to be dealt with. In literature, there are two prevailing approaches to tackle these challenges: (i) mathematical approaches that rely on modelling and optimisation as well as (ii) nature inspired approaches that rely on the advantages of swarm robotics.

Mathematical modelling based approaches offer tools that can be used to model systems and guarantee stability as well as robustness to disturbances. Leader-follower formations are one of the simplest multiple agent schemes for transporting loads and they are often a starting point for model based approaches [4–6]. A leader follower formation was used in [5] to transport a cable-suspended beam load by two aerial vehicles. This approach made use of mathematical modelling to derive a decentralised communication-less master-slave admittance controller. In their approach, the leader used implicit communication via cable forces to guide the system to a destination. A similar approach in [7] used visual feedback to maintain the relative position between the leader and the follower.

In [8], micro-UAV controllers were derived for transporting a prior known load configuration. An extensive system identification process, including a knowledge of the load's effect on cable tensions, was used in order to derive the parameters for the model. Furthermore, they relied on the use of globally available state data to inform the position of agents participating in the load transport exercise. This technique could be extended to more complex scenarios in which a load with an unknown mass distribution is to be transported. However, this would

\* Corresponding author.

E-mail addresses: [khuang5@sheffield.ac.uk](mailto:khuang5@sheffield.ac.uk) (K. Huang), [jchen118@sheffield.ac.uk](mailto:jchen118@sheffield.ac.uk) (J. Chen), [j.oyekan@sheffield.ac.uk](mailto:j.oyekan@sheffield.ac.uk) (J. Oyekan).



**Fig. 1.** A swarm transporting a load cooperatively by adapting to the load's mass.

require even more nontrivial complex modelling in order to guarantee the best micro-UAV configuration for transporting various loads [9]. In order to deal with different load masses of linear deformable objects (e.g cables, ropes), [10] applied the particle swarm optimisation and fuzzy logic in searching for the optimal parameters to apply to PID/PD controllers on micro-UAVs for a smooth, fast and reliable behaviour. Through a model of the system, the states of the cables and the required behaviour from the micro-UAVs was derived.

However, the modelling approach that couples micro-UAVs to loads makes it challenging to extend such systems to even more varying load scenarios and varying numbers of micro-UAVs. Towards expanding to larger numbers of micro-UAVs and generic loads, [11] modelled each subsystem (Quadcopter and load) independently with interactions between them represented as force and moment transfers. The advantage of this approach is that it can be scaled on demand to varying situations. Furthermore, their approach was able to ensure that the load mass was evenly distributed among the agents during transport. This was achieved through the use optimisation techniques to derive parameters for the PID controllers used in their scheme.

Even though the above approaches are susceptible to errors in modelling, when done correctly and with the right assumptions made, they are very robust to external disturbances. In order to tackle these modelling challenges, perhaps the use of machine learning techniques, such as neural networks and Reinforcement learning, could be used for system and controller identification purposes [12]. However, this introduces the need for extensive data collection and training.

In addition to stability and robustness, energy distribution and management among a team of micro-UAVs is important during collaborative transport. This is an area that still requires more detailed research [13]. The management of energy in multi-UAV collaborative transport is important since power failure in one of the micro-UAVs can result in the failure of the whole operation. This is even more important when considering mathematical based approaches since it is difficult to model unpredicted failures. This is an area where the concept of swarm robotics could support in deriving strategies to ensure mission continuation and completion even in the presence of a failed agent.

Swarm robotics approaches are inspired by nature and offers the advantages of natural systems such as robustness (ability to cope and adapt with the death or loss of individuals), scalability (ability to perform well with different group sizes) and flexibility (the ability to cope with variation in the environments) [14]. By extension, autonomous aerial swarms are expected to be more capable than a single large vehicle, offering significantly enhanced flexibility, adaptability, scalability, and robustness [15]. Such systems offer the advantage of being resilient and tolerant of noise in the sensors and the environment [16]. Furthermore, swarm robotics offers the promise of achieving the completion of complex tasks via simplistic agents that follow simple rules in their controllers. Nevertheless, engineering efficient swarm behaviours is still a big research challenge. Many tools have been proposed to deal with this challenge. They include the use of reinforcement learning, evolutionary principles and behaviour based architectures. The former two rely finding the optimal control policy in a large search space while the latter relies on designing and adding modular behaviours to an architecture [2,3].

Behaviour based architectures provide the ability to develop hybrid nature inspired algorithms and design swarm system effectively by adding the required ingredients (behaviours) needed to accomplish a goal [17]. By adding the ingredients as modules, each module can be separately debugged, tuned and its effect on the system analysed. They also provide the ability to achieve hierarchical architectures in which the modules can be organised into multiple control levels with each operating at varying granularities, levels of abstraction and time scales [2].

Chung et al. [18] highlighted the potential of applying hierarchical architectures to achieve well-founded, computationally-efficient and scalable algorithms for aerial swarms. Hierarchical architectures provide scaffolding for algorithms and can result in scalable systems that can achieve decentralized planning, reasoning, learning, and perception in the presence of uncertainties. Furthermore, such approaches have been found to be useful in both the machine learning and control fields for dealing with complexity and high dimensionality. Chung et al. [18] proposed that learning and decision-making architectures to provide aerial robot swarms with high levels of autonomy and flexibility will ensure reduced risk and cost as well as robust autonomous operations.

The behaviours for a behaviour based architecture system can be designed in any way deemed necessary by the designer to achieve a goal. For example, in [19] a set of basic functionalities (called elementary behaviours) were designed and combined in a priority order to attain complex tasks. In [20], fuzzy rules were used to develop the behaviours of a swarm required to perform exploration and navigation in an environment. Apart from the use of classical controllers such as PID or LQR, the application of fuzzy logic is another way to achieve to flocking behaviours on UAVs [21,22]. However, this requires an adequate knowledge of the domain in order to design effective fuzzy rules.

From the perspective of communication and control mode, there are mainly two modes, centralized and decentralized method, which require different control strategies. In the first communication mode, the control strategy requires a centralized master which is assigned to receive the states of all agents, process and transmit control commands. The main benefit of this approach is that developer can easily modify the controller from master. However, from the system point of view, this strategy is not robust because of the high reliance and heavy computation on the master.

In decentralized communication mode, agents can process their own commands thereby reducing the computation cost on the master [23]. Nevertheless, it is well known in literature that one of the major bottlenecks in decentralized algorithms is explicit communication. As the number of agents in the swarm increases, the communication network topology becomes more complex increasing the communication delays and packet losses. This drastically affects the performance of the systems relying on them. One of the reasons for this is due to the limited capability of the agents in the swarm and the need of the agents to have enough information to estimate their own states as well as others for the purpose of flocking. Undoubtedly, it puts forward a high requirement for individual performance especially when considering cheap and low computational cost agents. As a result, it becomes more advantageous to make use of a coordination system that limits the need for explicit communication and hence complexity [24].

Inspired by biology, previous work by Reynolds proposed a distributed behavioural approach in which flocking can be achieved without the need for centralized communication and explicit communication between agents [25]. The resulting decentralized swarm controller has been extended by numerous researchers where they have showed that even if the individuals cannot perceive the entire flock, a swarm consisting of self-organized agents can still move as a flock through local information interactions [26]. Taking inspiration from Reynold's work, explicit communication was not required to achieve a flocking behaviour in this work. Instead the visual perception of neighbours was applied in

a flocking behaviour generator to achieve the components of flocking namely cohesion, aggregation and heading alignment.

In [13], the challenges raised for multi-UAV cooperative transport were highlighted. This included the need to explore energy optimization and even energy distribution in multi-UAV collaborative transport systems, the need to have a mechanism to decide how many multi-rotors would be required to pick up a complex shaped objects based on their mass distribution and shape as well as ensuring stability during transport. Furthermore, [27] highlighted that a cooperative transport system that is safe, scalable and able to perform agilely using only robotic vision and local communication between neighbours is still yet to be found.

In this work, we present results that offer a roadmap towards providing a solution to the latter challenge identified in [27] while addressing the former challenges raised in [13]. In particular, our contribution lies in applying one of the tools in swarm engineering, a behaviour based subsumption architecture, to address these challenges. The novelty of our architecture is that it is able to achieve scalable decentralised multi-UAV load transportation without the need of explicit communication between agents. In our architecture, coordination between agents is achieved by using data from low cost monocular vision perception and cheap ultrasonic sensors on each agent. Our approach enables individual agents to search for a load in an environment, attach to it via suction cups and carry it to a desired destination cooperatively (Fig. 1). During attachment to the load, the agents are distributed according to the mass requirements of the load thereby ensuring that energy usage is uniformly distributed among the agents. Furthermore, we show that our approach is robust to disturbances thereby highlighting its ability to ensure stability during load transport.

The rest of paper is arranged as follows: In the next section, we discuss the structure of our architecture in detail, including relative localization, the behaviours used in the architecture, mechanisms of behaviour fusion and load mass adapting cooperative transportation. In Section 3, simulation results are presented to validate the robustness of our strategy as well as its energy distribution properties. Finally, we conclude and put forward some suggestions for future work in Section 4.

## 2. Methodology

In this section, we discuss the sensors and set up for the agents in the swarm as well as the proposed subsumption architecture used for decentralised cooperative transportation.

### 2.1. Single agent model

Each micro-UAV agent in the swarm is based on a Quadcopter. The Quadcopter is equipped as shown in Fig. 2. A green sphere is used for the identification of neighbours and self-localization within the swarm. Ultrasonic rangefinder and proximity sensors are used to obtain boundary environment information for obstacle avoidance while a monocular camera is arranged in such a way that it obtains a panoramic vision of the environment. In order to obtain the panoramic image, a convex mirror was set upon the single monocular camera to obtain a 360-degree vision. Furthermore, as shown in Fig. 3, the basic flight control of agents is realized, in level 0, by a cascade PID controller involving position, velocity and attitude controllers.

### 2.2. Subsumption architecture

In this work, a subsumption architecture is designed to implement a decentralised cooperative transport by a swarm of Quadcopter agents. Three different behaviours were chosen based on this goal: bacterium behaviour, flocking behaviour and obstacle avoidance behaviour. The bacterium behaviour was used for exploring the environment as well as providing exploitation based on the profile of the load. The flocking behaviour provided the mechanism for cooperation between agents while the obstacle avoidance behaviour enabled agents to detect and avoid

obstacles in their environment. The three different behaviours make up the **combined behaviour** module shown as ② in Fig. 3.

The subsumption system architecture was arranged in 4 layers. Level 0 contained the cascade PID controllers to provide attitude and altitude control for the Quadcopters. Levels 1 and 2 contain the three different behaviours mentioned above. These three behaviours are fused in the **velocity fusion** module (③ in Fig. 3). In level 3 (④ in Fig. 3), the **collective target tracking** is performed in order to maintain formation during the transportation of a load. The architecture also contains an **environment perception** (① in Fig. 3) module for perceiving the environment. The specifics of this architecture are explained in detail in the following sections below.

### 2.3. Low level controllers module

The low level controllers module is made up of a series of cascade PID controllers, control mixer, motor control as well as GPS and IMU sensors. The cascade PID controllers are responsible for low-level control of the Quadcopters. It is made up of the position PID controller, the velocity PID controller and the attitude PID controller. The position PID controller accepts the desired destination position for the load to be transported as well as the current position of the agent. The output is then used as a velocity set point for the velocity PID controller. The velocity PID controller outputs the desired yaw, roll and pitch angles which is then used by the attitude controller. The control mixer takes the calculated thrust for the altitude of the Quadcopter and Euler angle rate as input from the attitude PID controller. It then outputs four motor speeds for a Quadcopter. The changes in the condition of the Quadcopter is then detected by the GPS and IMU module for feedback to the cascade PID controllers.

### 2.4. Environment perception

The environment perception module represents all the abilities that an agent uses to obtain information from the external environment. Agents can obtain range information around themselves and height of the load by proximity sensors and an ultrasonic sensor respectively. This information feeds into the obstacle avoidance behaviour and bacterium behaviour respectively. Furthermore, we use a panoramic camera on each agent to obtain relative coordinates w.r.t. the camera coordinate of their neighbours in the swarm by tracking green sphere on the top of each neighbouring Quadcopter. The following sections discuss this in more detail.

#### 2.4.1. Object recognition for relative localization

Relative localization is a prerequisite for swarm clustering [25]. Most of the on-board relative localization techniques use a monocular camera and mounted distinguishable markers. For instance, Saska et al. used an on-board real-time reformative flood-fill algorithm and a simple circular pattern composed of concentric black and white circles of known diameters to achieve relative localization [28]. ArUco markers were used by [29] on the leader to obtain relative position while [30] realized leader-follower drone formation flying by equipping the leader with a unique colored ball. In the processed monocular vision image, the ball occupies a number of pixels that can be used to extract relative distance and bearing using the radius of the ball. This strategy is resilient to boundary partial occlusion, boundary object segment and image distortion at the edge [30].

In this work, a similar technique to [30] was applied. A green sphere was used as a significant marker to represent each agent. Since the plane projection of a sphere at all directions in space is approximately circular, even under the case of distortion, this approach produces more robust results. Histogram equalization and Gaussian spatial filtering were applied to the raw image to improve image contrast and immunity to noise jamming of images. After that, global thresholds from  $D_{\min}$  to

Panoramic camera	resolution: 1024x256
Ultrasonic sensor	max range: 20m
Proximity sensor	max range: 0.5m
UAV	mass: 520g size: 0.3 x 0.3 x 0.02m max speed: 1m/s processor: Linux built 1 GHz 32 bit ARM Cortex processor with 1 GB DDR2 200MHz RAM, 3 axis gyroscope, 3 axis accelerometer, 3 axis magnetometer, pressure sensor
Cable	size: 0.01 x 0.01 x 0.36m mass: 36g
Suction cup	maximum lifting: about 100g

Fig. 2. Overview of single agent model.

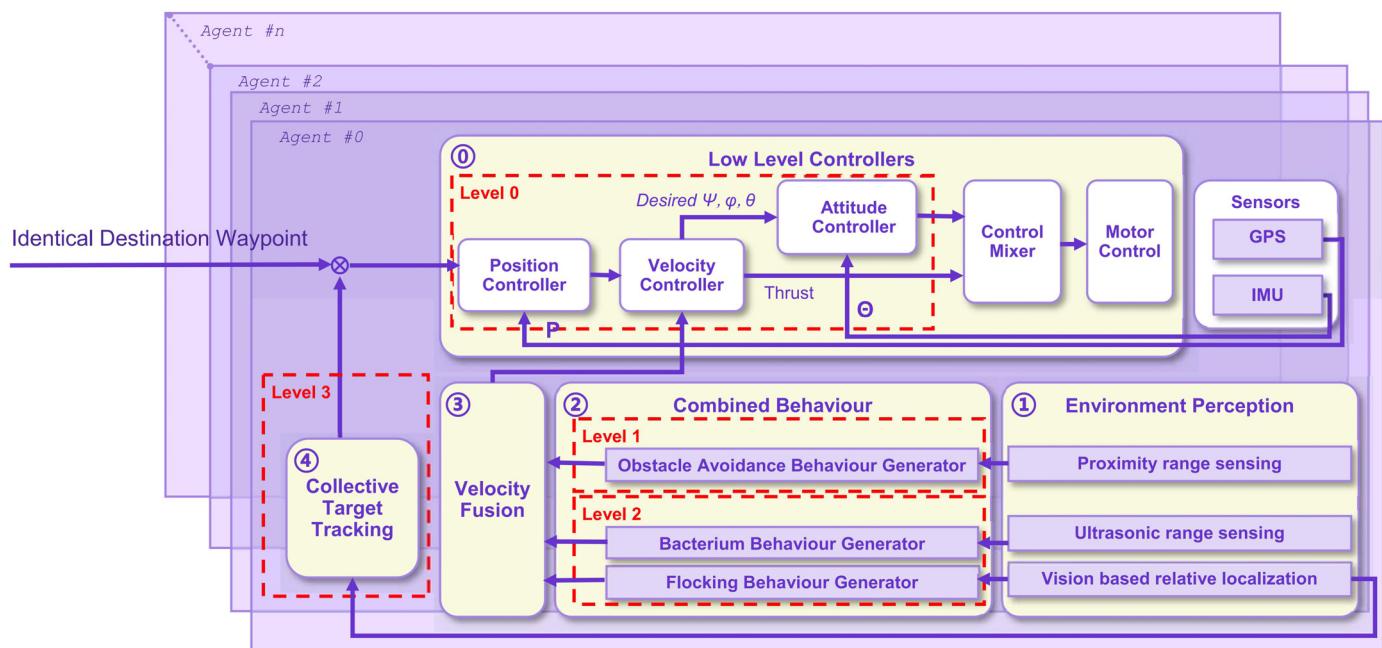


Fig. 3. Overview of subsumption architecture for each agent in the swarm.

$D_{\max}$  were used to achieve binarization and extract interested areas as in Eq. (1) where,  $I(x, y)$  is the processed image output.

$$I(x, y) = \begin{cases} 1, & \text{if } D_{\min} \leq I(x, y) < D_{\max} \\ 0, & \text{else} \end{cases} \quad (1)$$

Morphology processing was then applied to eliminate ambient light in-homogeneity. The above resulted in the detection of the ball in the monocular image from which the estimation of the relative positions of the neighbours was obtained.

#### 2.4.2. Bearing and distance estimation for relative localization

Extracting distance and bearing estimation to objects in the real world requires us to determine a relationship between a detected image and the real world.

An intuitive approach is to compare landmark radiiuses in images with that of the real world by applying generalized Hough Transform for geometric pattern detection [28]. However, this approach demands considerable on-board computation. Furthermore, the results of Hough Transform are extremely deterministic which means there is no compromise in testing whether patterns exist or not in an image. Thus, due to distortion and noise present in the camera, Hough Transform may give unstable results. This could negatively impact the inputs to the flocking algorithm resulting in destabilization of swarm formation. To avoid this, we used landmark average pixel intensity of connected regions to obtain the depth information of a target object. This approach provided immunity against noise [30,31].

By applying binarization as a prerequisite procedure for connected region detection, we were able to extract the average pixel intensity of the connected components as  $N_i$ . Using a speculative relationship  $F(*)$  between pixel intensity and depth information, we estimated the relative distance between agent  $n$  and agent  $i$ .

The relative bearing of neighbours from each agent  $n$  is estimated by the relative distance from the image plane centre of agent  $n$ , to the centroid position of the detected neighbour  $i$  on the image plane. Where  $(x^c, y^c)_n$  denotes the image plane centre of the  $n$ th agent, and  $(x_i, y_i)_n$  denotes the centroid position of detected neighbour  $i$  from the  $n$ th agent's perspective.

We establish a spherical coordinate system given by  $\rho_i = (\theta_i, \varphi_i, d_i)$  which represents the relative location of the  $i$ th neighbour in the current distributed agent  $i$ 's polar coordinate system.  $S_n$  is used to record local positions of all the neighbours of the  $n$ th drone where  $\theta$  is the relative pitch angle to a neighbour and  $\varphi$  is the relative yaw angle to a neighbour.

The above is captured in a relative localization mapping from image plane to the polar coordinate system as

$$\begin{aligned} S_n &= [\rho_1 \quad \rho_2 \quad \dots \quad \rho_i]_n = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_i \\ \varphi_1 & \varphi_2 & \dots & \varphi_i \\ d_1 & d_2 & \dots & d_i \end{bmatrix}_n \\ &= \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & F(*) \end{bmatrix} \begin{bmatrix} x_1 - x_c & x_2 - x_c & \dots & x_i - x_c \\ y_1 - y_c & y_2 - y_c & \dots & y_i - y_c \\ N_1 & N_2 & \dots & N_i \end{bmatrix}_n \\ \theta_i &\in [0, 2\pi], \varphi_i \in [-\frac{\pi}{2}, \frac{\pi}{2}], d_i \in \mathbb{R}^+ \end{aligned} \quad (2)$$

where

$$N_i = \sum_X \sum_Y I(x_i, y_i), \quad d_i = F(N_i) \quad (3)$$

where the  $K_x$  and  $K_y$  are constant parameters that depend on camera intrinsic parameters and as such affect the calculation of the relative bearing. The values of these parameters were determined empirically.

For the panoramic camera used, we chose  $\begin{pmatrix} K_x \\ K_y \end{pmatrix} = \begin{pmatrix} 2\pi/R_x \\ -\pi/2R_y \end{pmatrix}$ , where  $\begin{pmatrix} R_x \\ R_y \end{pmatrix}$  denotes the image resolution.

#### 2.5. Combined behaviour

In order to derive a swarm that searches and explores an unknown environment cooperatively, we propose that several behaviours should be running concurrently: (i) flocking behaviour for maintaining swarm formation, (ii) bacterium behaviour for searching and exploring the environment, and (iii) obstacle avoidance for avoiding collisions with the environment and other agents. As mentioned previously, the subsumption architecture provides a mechanism to derive a desired global behaviour through the fusion of several sub-component behaviours. The following sections describe the three chosen behaviours in detail and explain how their corresponding velocities were obtained in preparation for a fusion strategy in ③ of Fig. 3.

Furthermore, we subsume behaviours depending on conditions in the environment. For example, the bacterium behaviour is used for exploration and searching the environment. This behaviour is subsumed by the obstacle avoidance behaviour if a neighbour is detected in very close proximity. In this case, the obstacle avoidance behaviour causes the agent to accelerate away from the neighbour in order to avoid them. On the other hand, the agent should be unconcerned about neighbours that are far away since they have no immediate threat.

As a result, we split an agent's surrounding area into three zones (Fig. 4): near zone, mid zone and far zone. If the drone detects any threats in the near zone, i.e. other neighbours and obstacles, it will disregard any other behaviours and trigger obstacle avoidance behaviour directly. The near zone control is set as the highest priority layer in the subsumption architecture. Neighbours in the far zone are ignored partly because they are not an immediate threat and also because they are not detected due to the camera's resolution. The flocking behaviour is applied at the mid zone where neighbours begin to be detected by an agent.

##### 2.5.1. Flocking behaviour generator

The flocking behaviour of our swarm utilizes three rules. The three rules are separation, cohesion and friction which denote repulsion, aggregation and viscous friction-like interaction terms respectively [32,33]. The first two terms encourage agents to hold a certain distance from each other while avoiding collisions with each other. And the last term restrains attitude oscillations as the swarm seeks a steady state. We define a set of agents  $S_n^{mid}$ , that contain all the neighbours surrounding the  $n$ th agent in the mid zone:

$$S_n^{mid} = \left\{ \text{agents } i : i \neq n \wedge d_{\min} \leq \|d_i\| < d_{\max} \right\} \quad (4)$$

where  $d_{\min}$  and  $d_{\max}$  indicate the upper and lower bounds of the defined mid zone.

In order to simplify the calculation in every iteration, we introduce a virtual agent coordinate vector corresponding to the mid zone neighbours set  $S_n^{mid}$  of the  $n$ th agent [34]. Thus the flocking velocity component is given by

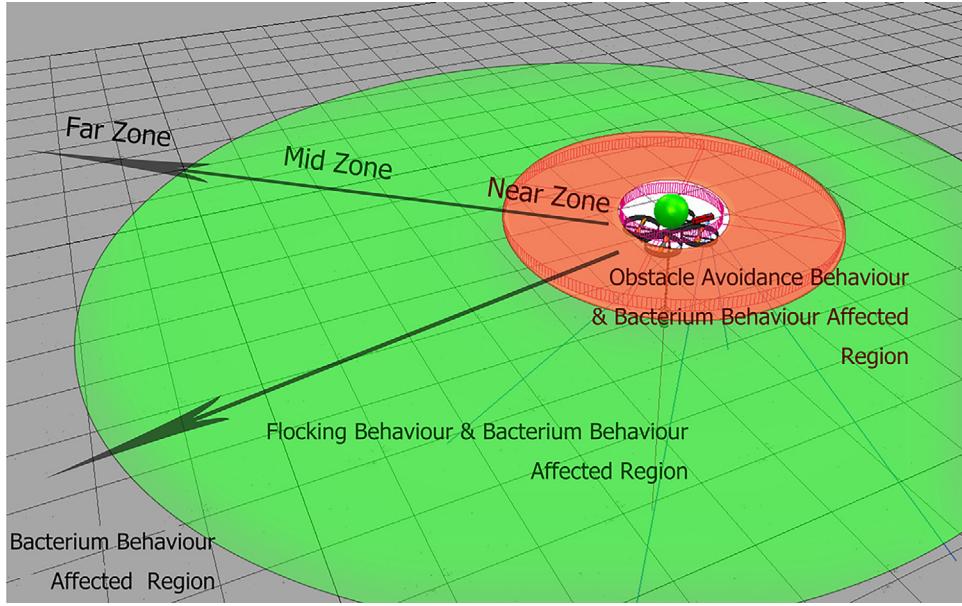
$$\begin{aligned} v_n^{sep} &= -k_{sep} \rho_n^{virtual} = -\frac{k_{sep}}{\|S_n^{mid}\|_0} \sum_{i \in S_n^{mid}} \frac{\rho_i}{\|\rho_i\|_2} \\ v_n^{coh} &= k_{coh} \rho_n^{virtual} = \frac{k_{coh}}{\|S_n^{mid}\|_0} \sum_{i \in S_n^{mid}} \rho_i \end{aligned} \quad (5)$$

where  $k_{sep}$  and  $k_{coh}$  are predefined constant separation and cohesion gains.  $\dot{\rho}$  is the approximate relative speed calculated by position difference between adjacent frames during relative localization of neighbours.

As mentioned previously, the velocity based friction term was introduced to dampen the vibrations of an agent attempting to reach a steady state rapidly.

This term is given by

$$v_n^{frict} = k_{frict} \rho_n^{virtual} = \frac{k_{frict}}{\|S_n^{mid}\|_0} \sum_{i \in S_n^{mid}} \dot{\rho}_i \quad (6)$$



**Fig. 4.** Zone definition with affected behaviour regions.

where  $k_{frict}$  is a friction gain that influences the intensity of the damping effect. Thus, the flocking term observed by swarm is

$$\mathbf{v}_n^f = \mathbf{v}_n^{sep} + \mathbf{v}_n^{coh} + \mathbf{v}_n^{frict} \quad (7)$$

Changing parameters  $k_{sep}$  and  $k_{coh}$  adjusts the flocking density of the swarm.

### 2.5.2. Bacterium behaviour generator

In an unknown environment, in order to find a target object and adapt to its density distribution, individuals in a swarm need to search and explore the unfamiliar area as well as apply a strategy to conform to the target object's distribution once found. Towards this, the chemotaxis behaviour in bacterium foraging serves as an inspiration. Bacterium foraging is an example of an adaptive composite search strategy that can explore an extensive region and perform priority based searches based on food profile distributions [35–37]. From an agent's perspective, the chemotaxis behaviour consist of tumble and run phases [38,39]. The tumble phase is a decision-making procedure in which the agent rotates itself and finds a random orientation for the next movement. The run phase is a migration process where the agent moves towards favourable gradients.

In this work, we apply the Berg and Brown mathematical model [40] to achieve bacterium foraging behaviour. This model describes the behaviour of a bacterium as it searches an area for food and adapt its search according to the concentration distribution of the food profile. This model has been successfully applied for the purpose of environment profile perceiving by Oyekan et al. [41] and is described as:

$$\tau = \tau_0 e^{\alpha \left( \frac{d\bar{P}_b}{dt} \right)} \quad (8)$$

$$\frac{d\bar{P}_b}{dt} = \tau_m^{-1} \int_{-\infty}^t \frac{dP_b}{dt'} e^{\frac{t'-t}{\tau_m}} dt' \quad (9)$$

$$\frac{dP_b}{dt} = \frac{k_d}{(k_d + C(x(t)))^2} \frac{dC}{dt} \quad (10)$$

$$|\mathbf{v}_n^b| = \frac{\beta_0}{\kappa_v \times C(x(t)) + 1} \quad (11)$$

where  $\tau$  is the mean run time, and  $\tau_0$  is the mean run time in the absence of concentration gradients.  $\tau_m$  is a time constant that presents a transitory memory for a single agent thus giving them knowledge about the

past conditions in the environment in the last  $\tau_m$ .  $P_b$  shows the fraction of the number of chemical particles bound to the bacterium's chemical receptor. Besides,  $\frac{dP_b}{dt}$  and  $\frac{d\bar{P}_b}{dt}$  represent the change of chemical perception with  $dt$  and the average change of chemical perception during  $\tau_m$  respectively. In this work, we use detected relative depth values of the load profile as  $C$ . The depth values were obtained by the ultrasonic sensor at the bottom of each Quadcopter.

$|\mathbf{v}_n^b|$  is the absolute bacterium behaviour velocity of an agent.  $\beta_0$  shows the highest speed of foraging for an agent and determines the basic velocity of the agent. An agent achieves the highest speed when it is searching an area without food, that is where concentration is equal to zero.  $\kappa_v$  is a constant and positive value, representing the amplification factor for food concentration and it is used to tune for the impact of food concentration.  $C(x(t))$  is the concentration at the current location and time varying from 0 to the infinity.

Upon analysing the equations above during run phases for an agent, the phase length is constantly changing over time since the concentration is changing. Equations (12) and (13) show that the first derivative of the run phase w.r.t time reduces and increases depending on the derivative of the receptor binding rate.

$$\frac{d\tau}{dt} = \tau_0 \alpha e^{\alpha \left( \frac{d\bar{P}_b}{dt} \right)} \times \frac{d}{dt} \left( \frac{d\bar{P}_b}{dt} \right) \quad (12)$$

$$\begin{aligned} \frac{d}{dt} \left( \frac{d\bar{P}_b}{dt} \right) &= \tau_m^{-1} \left[ \frac{dP_b}{dt} - \tau_m^{-1} \int_{-\infty}^t \frac{dP_b}{dt'} e^{\frac{t'-t}{\tau_m}} dt' \right] \\ &= \tau_m^{-1} \left( \frac{dP_b}{dt} - \frac{d\bar{P}_b}{dt} \right) \end{aligned} \quad (13)$$

In other words, the average value of the receptor binding rate tracks the current value of  $C$  and so the sign of  $\frac{d}{dt} \left( \frac{d\bar{P}_b}{dt} \right)$  is consistent with  $\frac{dP_b}{dt}$ . As a consequence, the agent will use a higher frequency of run phases if it is moving towards an optimal of the food profile ( $\nabla C(x(t))$ ) or initiate more tumble phases if it is moving in the opposite direction ( $-\nabla C(x(t))$ ).

### 2.5.3. Obstacle avoidance behaviour generator

In this work, we use artificial potential field method [42] to generate a repulsion velocity field for obstacle avoidance. By using *Force Inducing an Artificial Repulsion from the Surface*, the repulsion velocity increases significantly as the proximity with an obstacle decreases. This is shown

as

$$\nu_n^o = \mu \left( \frac{1}{\|\rho\|} - \frac{1}{\|\rho_0\|} \right) \frac{1}{\|\rho\|^2} \frac{\partial \rho}{\partial x}, \text{ if } \|\rho\| \leq \|\rho_0\| = d_{\min} \quad (14)$$

where  $\mu$  is a constant gain and  $\frac{\partial \rho}{\partial x}$  denotes the partial derivative vector of the distance from a point on an agent to an obstacle.

$$\frac{\partial \rho}{\partial x} = \begin{bmatrix} \frac{\partial \rho}{\partial x} & \frac{\partial \rho}{\partial y} & \frac{\partial \rho}{\partial z} \end{bmatrix}^T \quad (15)$$

In the near zone, there is only the obstacle avoidance behaviour acting. As a result, this causes agents to disperse rapidly without being restricted by the cohesion term in the flocking behaviour.

## 2.6. Velocity fusion

The Velocity Fusion module (③ in Fig. 3) was used to fuse the velocity values generated by the three behaviour algorithms, flocking, bacterium and obstacle avoidance. The output of the velocity fusion module acts on the velocity control loop of the agents. The main idea of this fusion algorithm is: (i) to provide a mechanism to tune the trade-off between behaviours respectively, (ii) to eliminate undesirable cluster driving forces in the swarm and (iii) to accelerate convergence while keeping the swarm's topology to match with the profile and mass distribution of the load.

This was achieved by summing up the three velocities as  $\nu_n^s$  according to

$$\nu_n^s = \nu_n^b + \nu_n^f + \nu_n^o \quad (16)$$

where  $\nu_n^b$ ,  $\nu_n^f$ , and  $\nu_n^o$  represents bacterium, flocking, and obstacle avoidance behaviour velocities respectively.

However, in this work, when there was a spacious environment and a comparatively small load, we observed that though part of the swarm found the load first, the agents can be pulled away from the load by other agents under the influence of the cohesive term in the flocking behaviour. This is especially true if the visible neighbours are operating on the edges of the detected load.

To address this, we introduce a parameter  $k_{flocking}$  (See Algorithm 1)

**Algorithm 1** Decentralised Load Searching and Mass Adapting Algorithm.

**Input:** The flocking behaviour velocity,  $\nu_n^f$ ; The bacterium behaviour velocity,  $\nu_n^b$ ; The obstacle avoidance behaviour velocity,  $\nu_n^o$ ; The current situations of proximity sensor and ultrasonic sensor,  $s_n^{prox}$ ,  $s_n^{ultra}$ ,  $s_n^{prox} = \text{True}$  means there are obstacles around the agent,  $s_n^{ultra} = \text{True}$  means agents have reached the target object;

**Output:** Fusion behaviour velocity,  $\nu_n^s$ ;

```

1: while True do
2:   if  $s_n^{prox}$  is True then
3:     Obstacle detected,  $\nu_n^s = \nu_n^o$ 
4:     goto final
5:   else if  $s_n^{ultra}$  is True then
6:     Target detected,  $\nu_n^s = \nu_n^b + k_{flocking} \nu_n^f$ 
7:   else
8:     Continue searching,  $\nu_n^s = \nu_n^b + \nu_n^f$ 
9:   end if
10:  final
11:  return  $\nu_n^s$ 
12: end while
```

which adjusts the impact of the flocking behaviour once a load is found. By using this strategy, high speed chemotactic agents operating on the boundary will have a reduced impact on agents who are already operating above the target load. Nevertheless, as members in a swarm, the agents operating on the load will still have an impact on their visible neighbours by attracting and pulling them into the load area with the cohesive term of the flocking behaviour. By following this strategy, the

time needed for converging on the load is reduced. The pseudo code for the searching strategy for every agent is described in Algorithm 1, where the parameter  $k_{flocking}$  has a range of between zero and one.

## 2.7. Collective target tracking and cooperative load transport to a location

The load to be transported is assumed to have the following properties:

- It is made up of a homogeneous material that can be detected by the sensing modalities on the Quadcopter. For example, if the sensing modality on the Quadcopter is a laser, then it is difficult for a laser to detect a load made of glass.
- The density of the material is uniform all over the load. Since  $Mass = Density * Volume$ , the mass of the sections of a load are determined by the volume and by consequence the shape or dimensions of the section.
- Using the last assumption, it means that the load to be transported can take any arbitrary shape as defined by the user and our scheme will work provided that the load shape can be detected by sensors on the Quadcopter for subsequent transport.

During collective transport, it is crucial to maintain formation in order to avoid collisions between agents. Due to the decentralized design of the flock, we assume each agent has a predefined destination coordinate stored onboard. We input this as a desired coordinate command directly into the position controller of the subsumption architecture of each agent.

As mentioned in the last subsection, the implemented mass adapting exploration obtains a distributed and optimized relative position for each agent that satisfies the mass distribution on the load. In order to reduce onboard computation cost, these relative positions are used by every agent as a reference signal to maintain their position in the swarm.

In the level 3 of subsumption architecture, these distinct relative positions and the specified destination coordinates are transferred to the collective target tracking module (④) on each agent for use to track and calculate the corresponding velocity commands.

## 3. Experimental results and discussion

In this section, we present the results of the performance of our architecture discussed in Section 2. The effectiveness of our strategy in cases of carrying various shapes of objects is evaluated in several simulation scenarios using six Quadcopters. It is important to emphasise that even though we used six team mates in our simulations, our proposed strategy can be scaled up to multiple team members. We examine how different parameters of our architecture's algorithm affect the load mass adapting strategy (non-uniform swarm network) and compare its performance with non mass adapting strategy (uniform topology swarm network). In order to compare both strategies, we study their robustness to noise as well as their energy consumption during operation.

The simulations are run in the robot simulator CoppeliaSim and ROS environment (Robot Operating System). The Quadcopters used in the swarm are based on the AR.drone2.0 designed by Parrot and can be found in CoppeliaSim. Additional grasping mechanism, visual sensor module, proximity sensor module, ultrasonic sensor and a significant landmark for identification are added to each Quadcopter. The spherical vision sensor mounted on the Quadcopters provide raw images of 1024×256 pixels. The proximity sensor can detect 0.5m range around it to avoid collision while the ultrasonic sensor has a maximum sensing range of 20 meters and provides the relative height of objects below the agent.

In the actual design, the Quadcopter is built of carbon fiber tube and weighs approximately 520g. The onboard autopilot is custom design including Linux built 1 GHz 32 bit ARM Cortex processor with 800MHz video DSP, 1 GB DDR2 200MHz RAM, 3 axis gyroscope, 3 axis accelerometer, 3 axis magnetometer, pressure sensor and ultrasound

sensors. Our distributed controllers communicate with CoppeliaSim via ROS by sending propeller rotation speeds to rotors for control.

Our results and analysis in this section is mainly divided into three parts: performance of convergence onto various load profiles, cooperative transport of loads and evaluation of robustness of our strategy.

### 3.1. Converging on load objects

#### 3.1.1. Converging on various load profiles

In order to test the performance of our approach to converge onto various load profiles, we selected several representative shapes including convex and concave. The shapes were rectangle, peanut and the letter “L” shape. The purpose here was to establish that our proposed strategy was capable of converging onto various load shapes. The results are shown in Fig. 5. In all the scenarios, robots begin with completely random positions in an unknown environment. Under the action of bacterium chemotaxis and flocking behaviour, the swarm finds and adapts to the mass distribution of the detected load.

In order to measure the adaptability of the swarm to the load mass distribution, the Kullback Leiber value, which is popularly applied in estimating similarity between two distributions [41], is proposed as criteria. The lower the KL divergence value, the closer the distribution of the swarm is to the mass distribution profile of the load.

$$J_{KL} = - \sum_{Y}^A \sum_{X}^A S(x, y) \log \left( \frac{F(x, y)}{S(x, y)} \right) \quad (17)$$

$S(x, y)$  represents the load mass distribution in a certain rectangular area  $A(X, Y)$ , that is slightly larger than the object;  $F(x, y)$  denotes the normalized population density distribution of the swarm, achieved by calculating the number of drones per meter around each location in  $A(X, Y)$ . At the beginning of a simulation run, the KL distance is large. This is because the aerial robots in the swarm are randomly placed in the environment resulting in differences between the swarm population distribution and the target load mass distribution. As seen in Fig. 5, during simulations, the swarm distribution converges rapidly to the load mass profile, and eventually maintains a small value. The results confirm that our strategy can be applied to various load profiles and not limited to only convex objects.

#### 3.1.2. Converging with different gains

For complex load profiles with irregular surfaces, the mass adapting topology of the swarm can be adjusted by changing the parameters of our algorithm. For example, as discussed in Section 2.6, our Algorithm 1 implements a  $k_{flocking}$  gain in line 6. This gain can be used to adjust the impact of the flocking behaviour once a load is found.

Figure 6 highlights the comparison results of when  $k_{flocking} = 1$  (that is no  $k_{flocking}$  adjustment when the load is found) versus  $k_{flocking} = 0$  (that is  $k_{flocking}$  adjustment when the load is found) when applied to a rectangular target load. As seen in the results, the approach of adjusting  $k_{flocking}$  when a load is found is effective in increasing the convergence rate to the load's profile. It is also effective at adapting more closely to the profile of the load.

In order to further investigate the effects of varying  $k_{flocking}$  gains on the mass adapting swarm topology and the  $KL$  divergence values, we used a peanut shape target load. Initially, flock members are randomly distributed in the environment to start searching and exploring for the load. This results in an initial  $KL$  divergence value 25. We tested  $k_{flocking}$  gains of 0, 0.05 and 0.15 with results in Fig. 7 showing that as  $k_{flocking}$  gain increases, the convergence rate increases and the  $KL$  divergence cost also increases. The effect of increasing the  $k_{flocking}$  gain can be seen more visually in Fig. 8. It can be seen that the swarm's topology adapts more closely to the peanut shape when the  $k_{flocking}$  gain is set to 0.

It should be noted that even when the  $k_{flocking}$  gain is set to 0 and nullifies the flocking velocity behaviour, the obstacle avoidance behaviour will be activated if a collision between agents is imminent. The obstacle

avoidance behaviour prevents collision and leads to separation between the agents. The obstacle avoidance behaviour is also instrumental in repelling agents to explore other areas on the load while the bacterium behaviour drives agents to individually find an optimal on the load profile leading to the aggregation of agents. Furthermore, the movement of the agents is influenced by the profile of the load leading to mass adapting (see Eq. (10)). As a result, a  $k_{flocking}$  gain of 0 leads to lower  $KL$  divergence values.

The combination of the effects of aggregation (provided by bacterium behaviour) and separation (provided by the obstacle avoidance behaviour) causes a form of flocking to occur over the load profile even though the  $k_{flocking}$  gain of the flocking behaviour is set to 0. Nevertheless, this does not mean that the flocking behaviour has no relevance when adapting to the profile of a load. It is instrumental in recruiting other agents to the load once found as well as providing structure to the swarm topology in the presence of the random nature of the bacterium behaviour. As such, having the ability to tune the  $k_{flocking}$  gain ensures that we have some level of control over the swarm structure.

### 3.2. Cooperative transportation

In the last section, we presented the effectiveness of Algorithm 1 in converging to the profile of a load whereas in this section, the results of the cooperative transport of the load are presented. In order to demonstrate the necessity and advantage of applying load mass adapting swarm topology for cooperative transport, we compare the results of two sets of experiments. In the first experiment, a uniform swarm topology is used to transport the load while an load mass adapting topology is used in the second experiment.

#### 3.2.1. Load collaborative transport

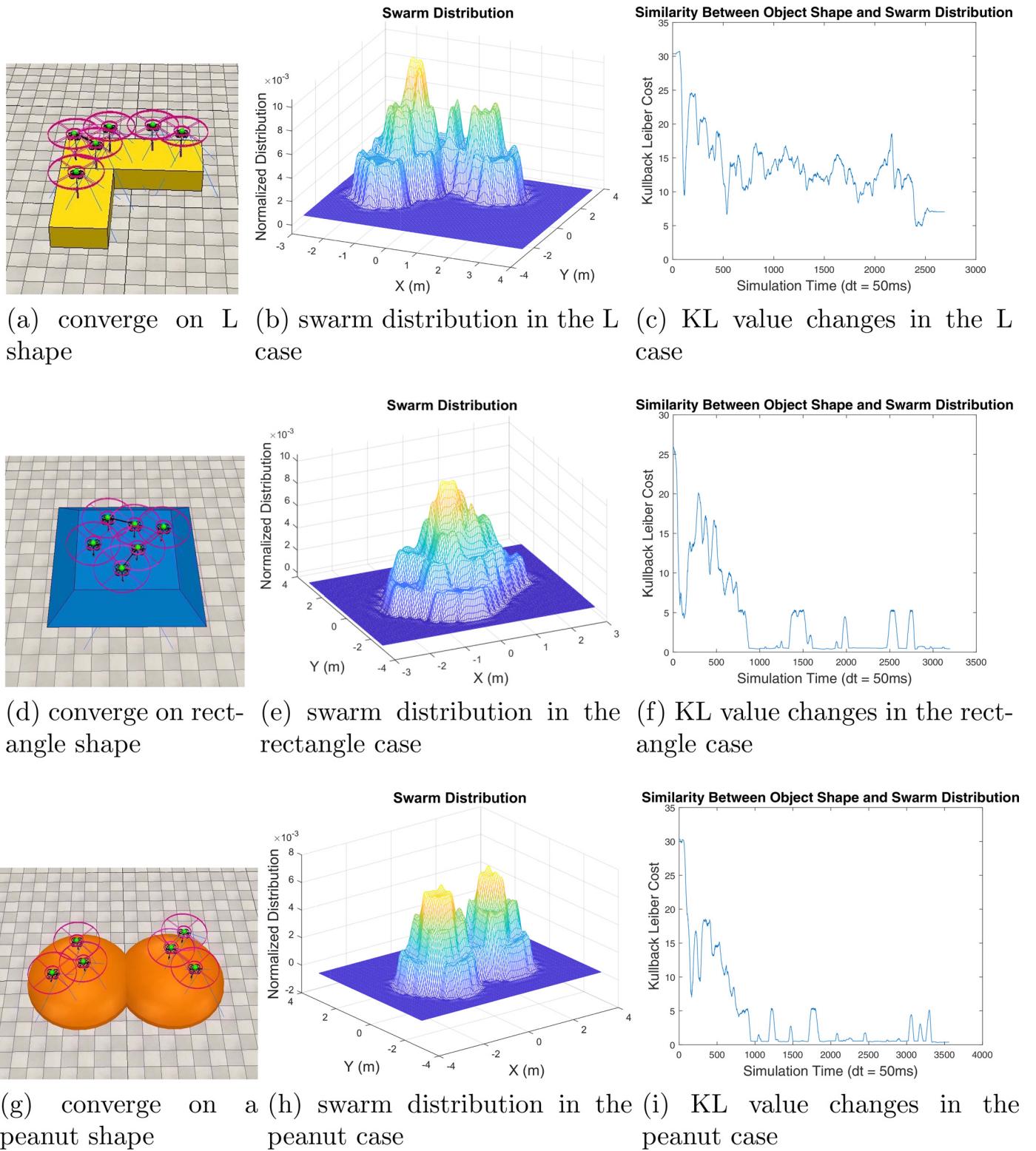
As discussed in the methodology section, the swarm executes collective tracking in which each agent maintains their relative distance to each other. At the beginning of the simulations, agents are placed randomly in a simulated square arena that has a dimension of 20meters by 20meters. After converging to the load, the agents descend and attach to the load via a vacuum suction cap held at the end of a cable. The load is then lifted to 1.3 meters above the ground. Subsequently, the swarm performs the collective tracking operation by carrying the load to the destination (Fig. 9) where the payload is released.

The individual positions of the agent in the swarm and hence the topology distribution when they converge on the load is obtained by Algorithm 1 prior to load attachment. After attachment to the load, the agents collaboratively transport the load to a desired coordinate. Figure 10 shows the trajectories for the agents executing Algorithm 1 prior to attachment to a peanut and a rectangle shaped load as well as during the collaborative load transport to a coordinate. The ability of the collective tracking module (@ in Fig. 3) in keeping agents at their relative positions in the swarm can be seen in Fig. 10. Figure 9 also shows the swarm topology during collaborative load transport to the desired coordinate position. It should be noted that the collective tracking module maintains the relative position between agents without communication between the agents.

#### 3.2.2. Energy comparison

In this section, we discuss how a mass adapting swarm topology impacts the energy consumption of individuals in the swarm. Towards this, we make use of an uneven load as shown in Fig. 11. We tested three scenarios: (i) uniform swarm topology with power restrictions placed on the agent; (ii) uniform swarm topology without power restrictions placed on the agent and (iii) mass adapting topology with power restrictions placed on the agents. We shall now discuss each scenario in turn.

**Uniform swarm topology with power restrictions placed on the agent:** We commanded the agents to lift a load to an altitude of 1.3m as seen in Fig. 11. It is seen in Figs. 11a and 12 b that the agents could



**Fig. 5.** Converging of the swarm to various shapes where  $K_{flocking} = 0$ .

not lift the load to the desired altitude. The energy consumption of each of the agents is shown in Fig. 12a where it can be seen that the power usage of each agent varies depending on where they are attached on the load. In this experiment, we restricted the amount of thrust that each motor on the drone can generate. The limitation was informed by the physical parameters of the Parrot Drone. Two important power values,

no-load and maximum power, were used during our analysis. The no-load power represents the maximum power required by agents to hover steadily without any payload, while the maximum power is the upper limit on the motors. It can be seen in Fig. 12a that  $Drone_0$  and  $Drone_1$  reached the maximum power while  $Drone_4$  and  $Drone_5$  were below the no-load power value.

## Similarity Between Object Shape and Swarm Distribution

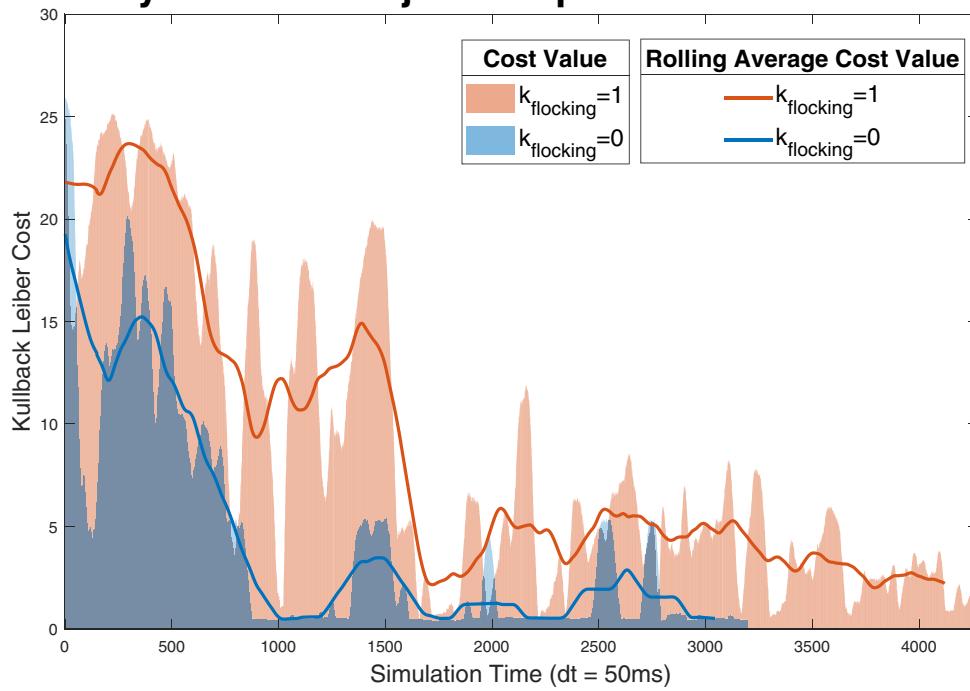


Fig. 6. Converging comparison of changing the parameter  $k_{flocking}$  from 1 to 0 once the rectangular load is found.

## Converging Comparison

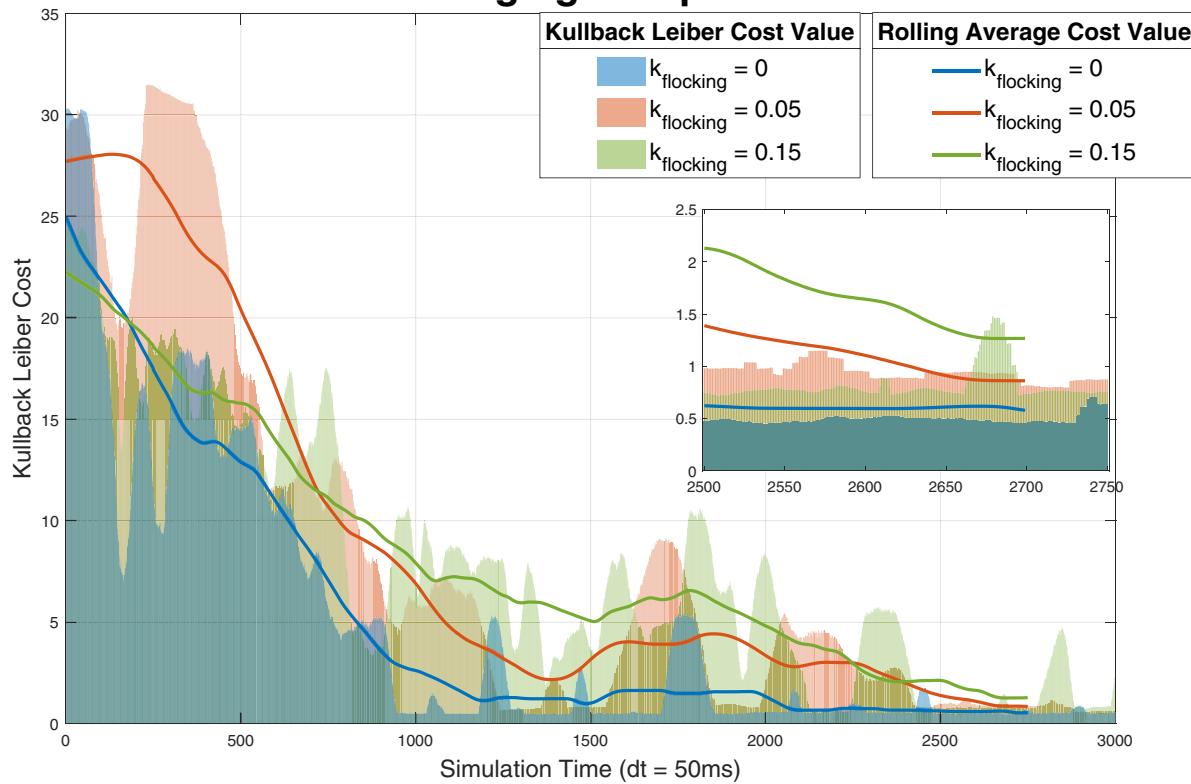
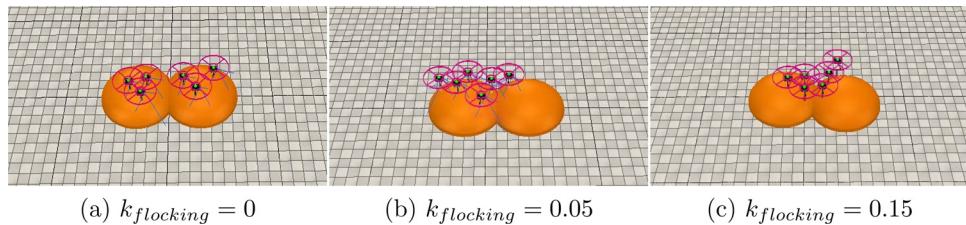
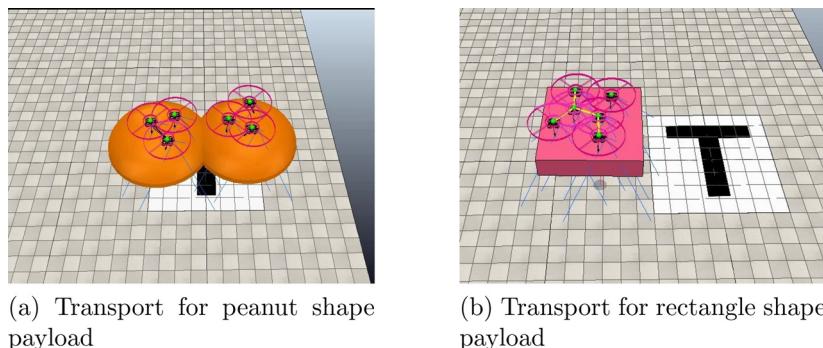


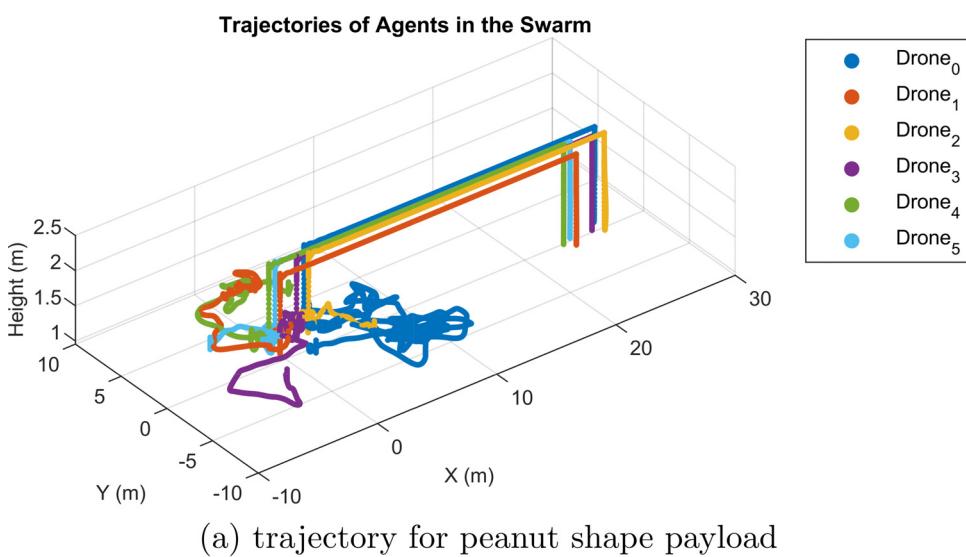
Fig. 7. Converging comparison under various  $k_{flocking}$  for a peanut shape load.



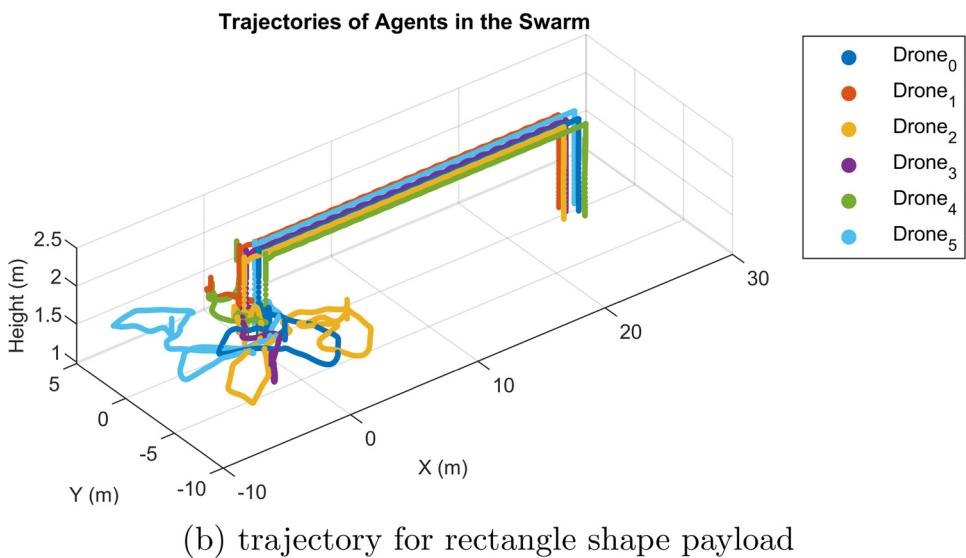
**Fig. 8.** Converging performance on a peanut shape load, under various  $k_{flocking}$  values.

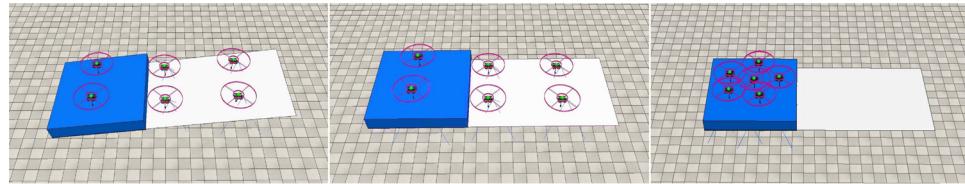


**Fig. 9.** Cooperative transportation of various loads to a destination.

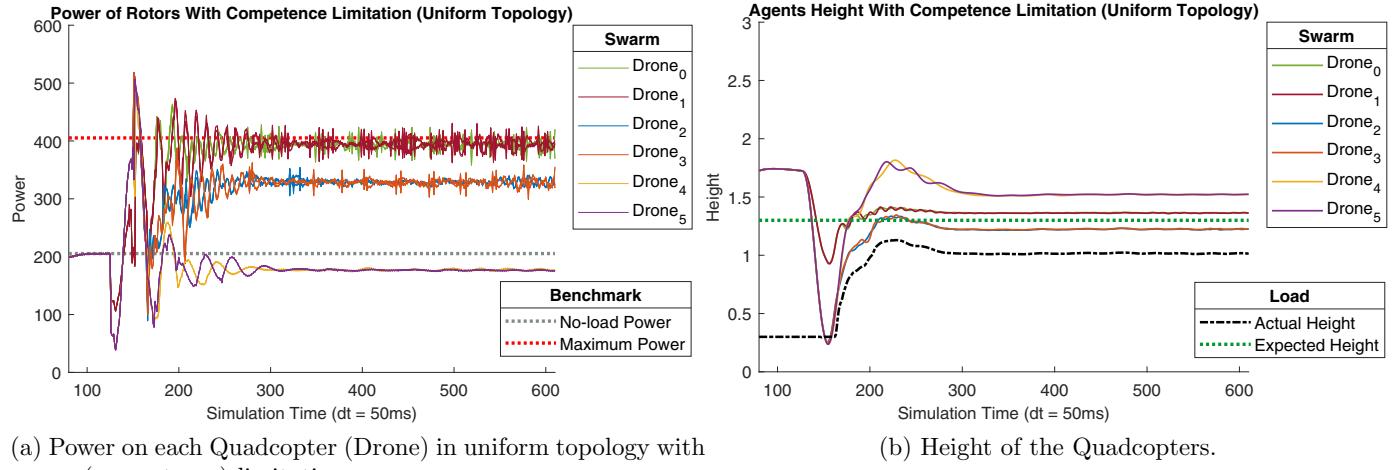


**Fig. 10.** Cooperative transportation trajectory.





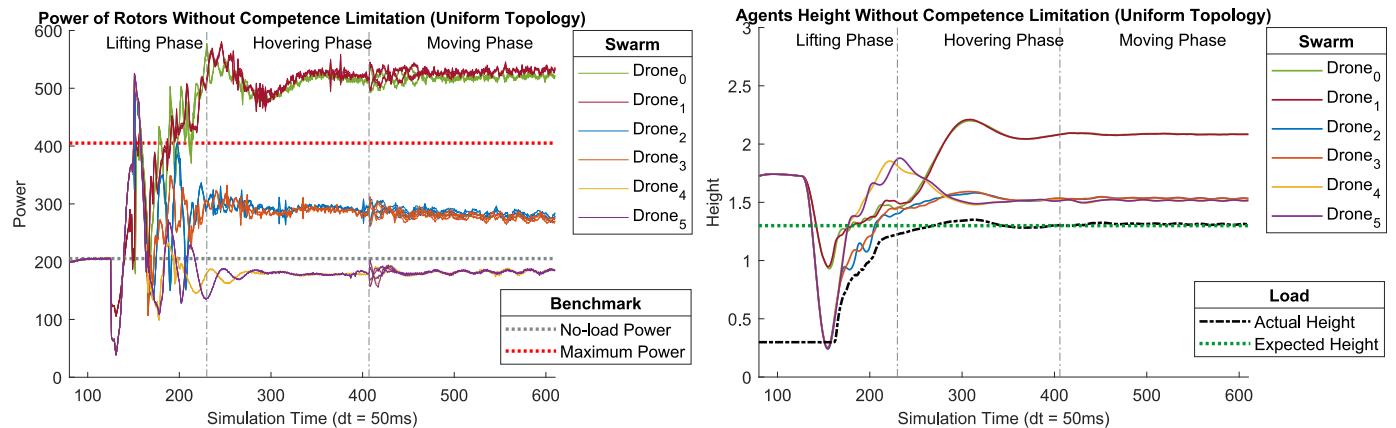
(a) uniform topology with power limitation      (b) uniform topology without power limitation      (c) mass adapting topology



(a) Power on each Quadcopter (Drone) in uniform topology with power (competence) limitation.

(b) Height of the Quadcopters.

**Fig. 12.** In Uniform topology, power on each Quadcopter varies depending on where they are attached to the load. The Quadcopters are not able to lift the load.



(a) Power on each Quadcopter (Drone) in uniform topology without competence limitation

(b) Height of the Quadcopters when there is no competence limitation.

**Fig. 13.** Power on each Quadcopter and their height in uniform topology scenario. Energy distribution is uneven and the Quadcopters lift the load.

**Uniform swarm topology without power restrictions placed on the agent:** By removing the power limitations on the motors of each agent, it can be seen in Fig. 11b that the agents were able to lift the load to the desired height. However, to achieve this, the power usage of two of the drones (*Drone<sub>0</sub>* and *Drone<sub>1</sub>*) had to unrealistically go above the maximum power capabilities of the motor (Fig. 13). In this case, these two drones were doing majority of the heavy lifting.

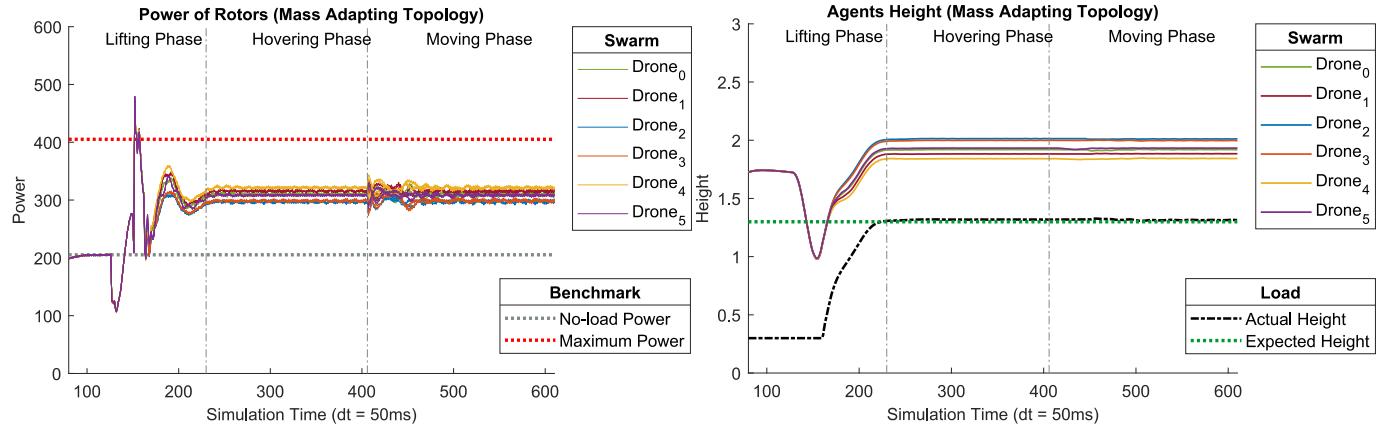
**Mass adapting topology with power restrictions placed on the agents:** We then deployed our mass adapting strategy as seen in Fig. 11c. With our strategy, the agents were able to lift the object with a more uniform power usage that is below the maximum power limitation of the motors on each agent (Fig. 14). When considering physical wear and tear, this approach is more likely to be gentler on the motors as well

as the batteries of each agent. Furthermore, by comparing the power usage of the mass adapting swarm topology versus the uniform swarm topology without power limitation, it was discovered that the power usage of the mass adapting topology is lower (Fig. 15). From the above results, it can be concluded that uniform distribution formation demands a higher energy usage from the swarm.

### 3.2.3. Robustness comparison

We then analysed the stability of the mass adapting swarm topology to external disturbances. In order to introduce disturbances, we make use of a simulated wind tunnel in CoppeliaSim.

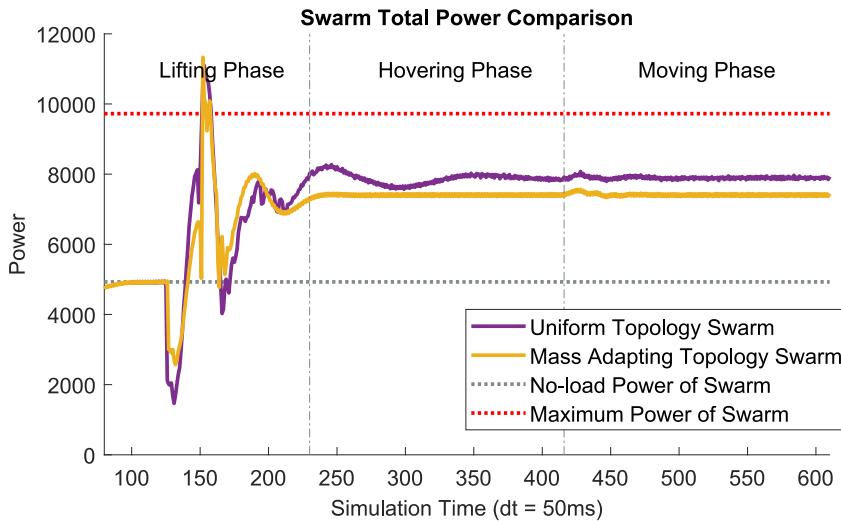
The wind from the wind tunnel is simulated by small particles that are generated randomly and propelled from the wind tunnel. The swarm



(a) Power on each Quadcopter (Drone) in mass adapting topology

(b) Height of each Quadcopter.

**Fig. 14.** Power on each Quadcopter and height in load mass adapting topology scenario. Energy distribution among the members is even and the Quadcopters lift the load.



**Fig. 15.** Power comparison between a uniform topology swarm and a mass adapting topology swarm.

system is then placed in the wind tunnel (Fig. 16) where a steady disturbance is exerted on it. The system's robustness can then be analysed and inferred by observing the sensitivity of the entire system to the external disturbances [43]. In the previous works, the stability of a load carried by agents is analysed via precise dynamic modelling due to the load being carried at pre-defined and known connections [44,45]. In our case however, the swarm formation and the points of connection to a load is unknown prior to the load being lifted. As a result, we use kinetic energy of the load as a metric to infer the robustness of the swarm system to disturbance. The kinetic energy of the load was obtained according to:

$$E_{\text{translation}} = \frac{1}{2}mv^2, \quad E_{\text{rotation}} = \frac{1}{2}J\omega^2, \quad E_{\text{kinetic}} = E_{\text{translation}} + E_{\text{rotation}} \quad (18)$$

where  $m$  and  $J$  are the mass and the moment of inertia of the load;  $v$  and  $\omega$  are translation and rotation velocities of the load. A higher load kinetic energy means that the robustness of the system is low and vice versa.

In order to further investigate the robustness of the mass adapting swarm system, we tested how cable length and cable angle affects the kinetic energy of the load when it is lifted. Towards this, we introduce a scaling factor that expands and contracts the swarm by increasing and

reducing the distance between individual members respectively. Firstly, the swarm lifts the load and then expands by a scaling value so that the cable angle can increase while keeping the swarm's distribution according to mass.

As a result of the above, the cable length and swarm scaling values are selected as independent experimental variables while the kinetic energy of the load is the dependent variable (Fig. 17). The experiments begin from a static state, after which disturbance is added for a period of time (60s) and the kinetic energy of the load obtained.

We also define two swarm scenarios: dynamic and non-dynamic. The dynamic scenario is a situation in which the swarm is influenced by the load's oscillation and corresponding vibrations in the cable when the system is under external disturbance. The non-dynamic scenario is an ideal situation, where the agents in the swarm are not affected by the vibrations in the cables and the oscillations of the load. We conducted experiments to compare the influences of cable length and cable angles (through swarm scaling) in these scenarios in order to derive the key principles for achieving load stability during transport.

**Experimenting with swarm scales:** In the experiments of how swarm scales affect the robustness, dynamic and non-dynamic swarm scenarios are loaded into the wind tunnel respectively with various scaling values. As shown in Fig. 18a, the robustness of the system is reduced

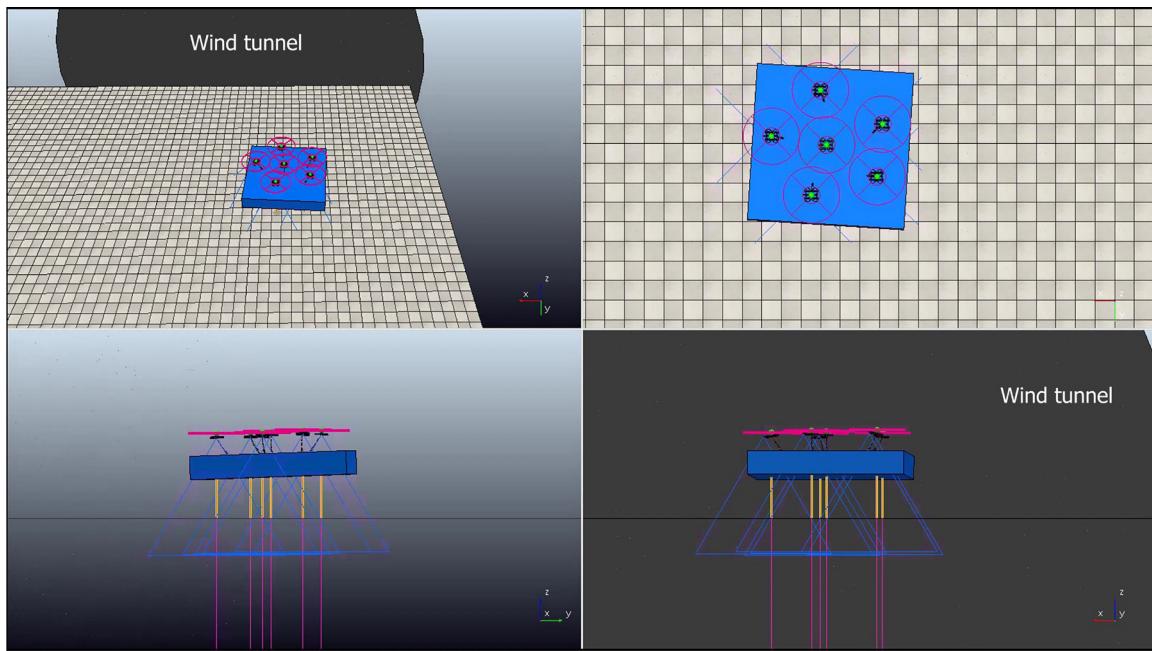


Fig. 16. Robustness evaluation by wind tunnel disturbance.

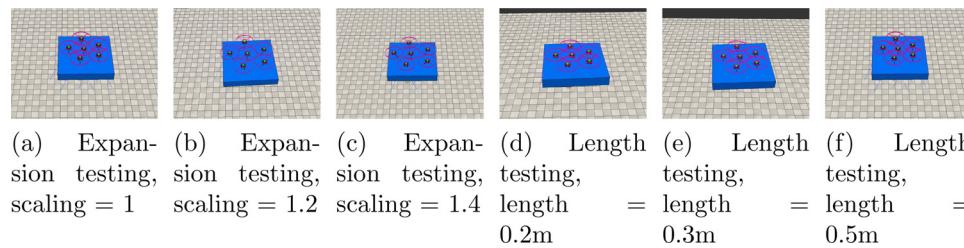


Fig. 17. How cable lengths and swarm scales influence system stability.

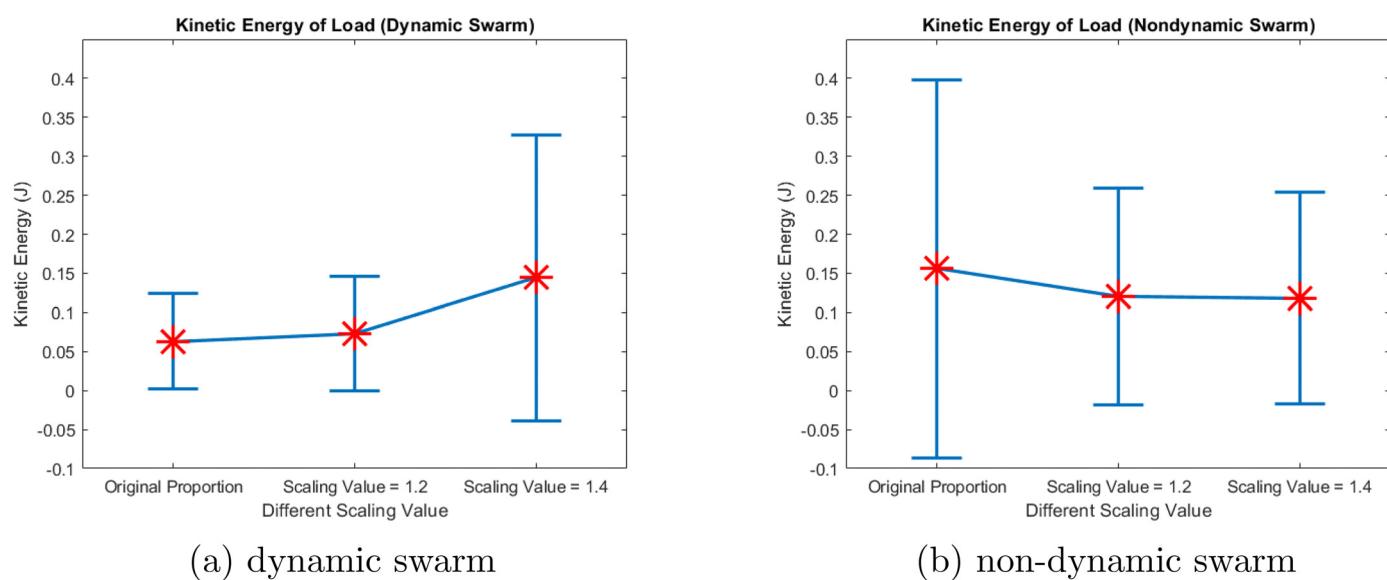
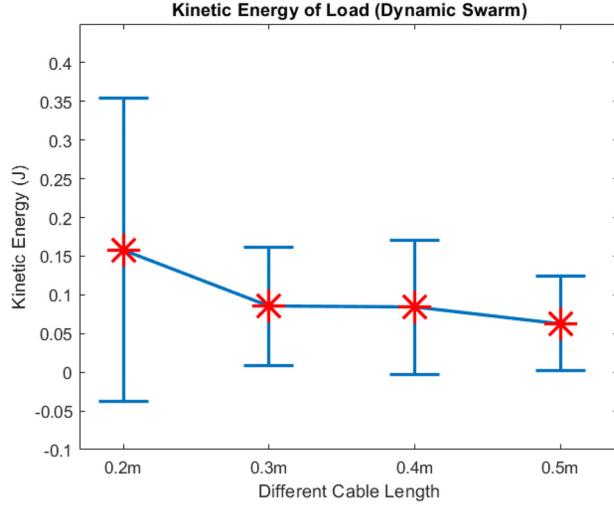
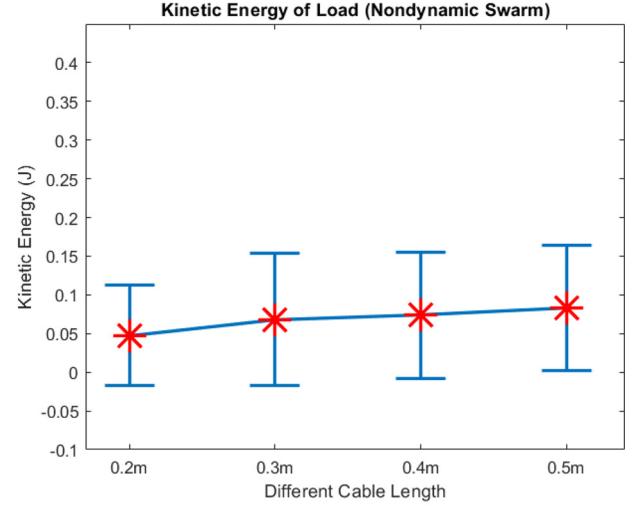


Fig. 18. Robustness comparison of different scales in dynamic swarm and non-dynamic swarm. The red stars in the error bars show the average kinetic energy while the blue bars show the variance of the kinetic energy over 60s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

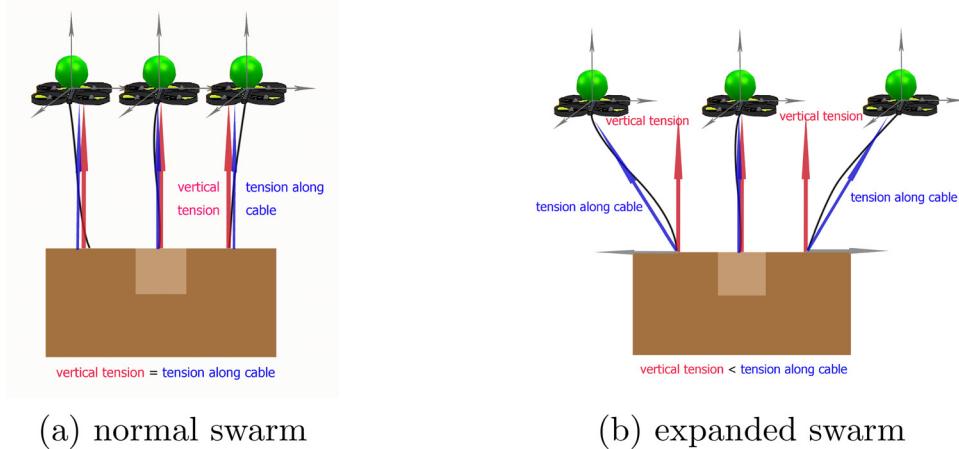


(a) dynamic swarm



(b) non-dynamic swarm

**Fig. 19.** Robustness comparison of different cable lengths in dynamic swarm and non-dynamic swarm. The red stars in the error bars show the average kinetic energy while the blue bars show the variance of the kinetic energy over 60s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 20.** Visual representation of how swarm scale affects robustness.

with the expansion of the swarm in the dynamic scenario. However, we observe the opposite results in non-dynamic scenario, where the robustness is improved by the expansion of the swarm (Fig. 18b).

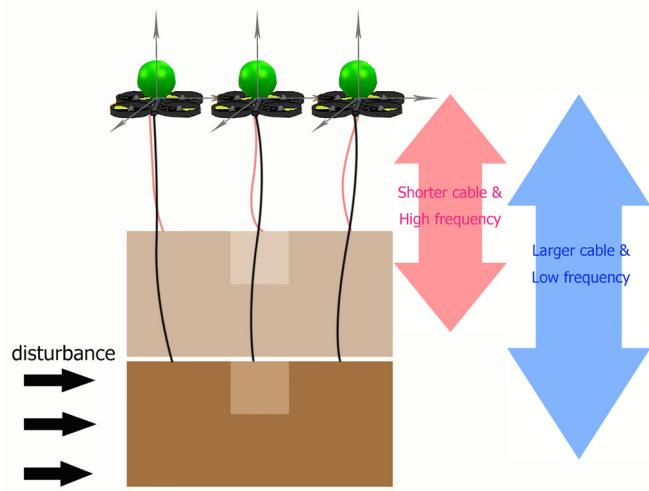
The reason for these results is that the larger values of the scaling factor leads to larger angles between the cables and the load. This increases the tension on the cables (Fig. 20) which dramatically affects the Quadcopters attitude in the dynamic model and makes them unstable. While in the non-dynamic model, the Quadcopters are consistently stable, so that the load is pulled and holds steadily.

**Experimenting with cable lengths:** In the dynamic scenario, we observed that the load-suspended system has less kinetic energy and is hence more robust when a longer cable is used (Fig. 19a). When the load is affected by disturbances, its motion characteristics is oscillatory and similar to that of a simple pendulum. The oscillation frequency is higher in a system with shorter rope lengths. In the dynamic scenario, this requires the drone to make demanding fast attitude adjustments; whereas with longer cables, low-frequency oscillations occur and the load has lower kinetic energy or better robustness (Fig. 21). However, in the non-dynamic scenario, the opposite results are observed (Fig. 19b). As cable length increases, the kinetic energy of the load increases.

#### 4. Conclusion and future work

Ensuring stability, robustness and even energy distribution among a team of Quadcopters are some of the pressing challenges during cooperative transport of complex loads [13]. Inspired by swarm robotics, we have presented a behaviour based subsumption architecture that achieves load mass adaptive cooperative transport by a swarm of Quadcopters. Our strategy is able to ensure robustness to noise and even energy distribution between agents. Furthermore, our architecture is able to accelerate the converging to various load mass distributions without prior knowledge.

The subsumption architecture is based on the combination of three behaviours: flocking, bacterium and obstacle avoidance. The choice of these behaviours was based on the desired goal of the swarm. The flocking behaviour enabled agents to move cooperatively while avoiding collisions with each other. The bacterium behaviour provided agents with the capability to explore an environment in search for the desired load. Once the load is found, the bacterium behaviour enables the agents to adapt their motions based on the profile of the desired load. The obstacle avoidance behaviour prevents collisions from happening with objects in the environment including the agents themselves.



**Fig. 21.** Visual representation of how cable length affects robustness.

The resulting swarm is decentralized and does not use explicit communication between members. Instead it uses vision as well as proximity sensors to achieve cohesion, separation and obstacle avoidance. The swarm is able to adapt to the profile (mass distribution) of a detected load resulting in a mass adapting swarm topology. By applying a suction mechanism, the swarm is able to grab the load and carry it to a target location while maintaining the load profile topology. Through simulation results, we demonstrate that our load mass adapting approach conserves more energy during transport than an approach in which the swarm maintains a uniform topology. During our experiments, we discovered that if the load's profile is smaller than the footprint of the swarm, some members of the swarm will be repelled away by members already on the load. The repelled members then keep exploring the environment for other loads. This highlights the potential for our approach to automatically assign the appropriate number of individuals to a load's transport depending on the need. This will be explored further in future work.

Furthermore, we show how the length of cables used as well as the cable angle between the load and the agent impacts swarm system robustness during load transport. Through the use of a kinetic energy metric, it was discovered that longer cable lengths and smaller angles improve robustness of the swarm system.

Since in this work, we assumed a load of uniform density throughout (that is a load made of a homogeneous material through-out), future work should look into how loads of varying densities (heterogeneous materials) can be effectively transported using a swarm of cooperative agents. Furthermore, investigations should be carried out into how real time dynamic adjustments of cable lengths during disturbances could further improve swarm stability and robustness. This is particularly important when dynamic load shifting during transport (e.g transporting fluids) is taken into consideration.

#### Declaration of Competing Interest

None.

#### CRediT authorship contribution statement

**Kangyao Huang:** Formal analysis, Methodology, Data curation, Software, Writing – original draft, Visualization. **Jingyu Chen:** Writing – original draft, Resources, Software. **John Oyekan:** Project administration, Conceptualization, Supervision, Validation, Writing – review & editing.

#### References

- [1] B.E. Jackson, T.A. Howell, K. Shah, M. Schwager, Z. Manchester, Scalable cooperative transport of cable-suspended loads with UAVs using distributed trajectory optimization, *IEEE Rob. Autom. Lett.* 5 (2) (2020) 3368–3374, doi:[10.1109/LRA.2020.2975956](https://doi.org/10.1109/LRA.2020.2975956).
- [2] S.J. Chung, A.A. Paranjape, P. Dames, S. Shen, V. Kumar, A survey on aerial swarm robotics, *IEEE Trans. Rob.* 34 (4) (2018) 837–855, doi:[10.1109/TRO.2018.2857475](https://doi.org/10.1109/TRO.2018.2857475).
- [3] F.Y. Hadaegh, S.J. Chung, H.M. Manohara, On development of 100-gram-class spacecraft for swarm applications, *IEEE Syst. J.* 10 (2) (2016) 673–684, doi:[10.1109/JSYST.2014.2327972](https://doi.org/10.1109/JSYST.2014.2327972).
- [4] A. Tagliabue, M. Kamel, S. Verling, R. Siegwart, J. Nieto, Collaborative transportation using MAVs via passive force control, in: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 5766–5773.
- [5] M. Tognon, C. GABELLIERI, L. Pallottino, A. Franchi, Aerial co-manipulation with cables: the role of internal force for equilibria, stability, and passivity, *IEEE Rob. Autom. Lett.* 3 (3) (2018) 2577–2583.
- [6] P.-X. Wu, H.-A. Hung, C.-C. Yang, T.-H. Cheng, Cooperative transportation of drones without inter-agent communication, in: *2020 American Control Conference (ACC)*, IEEE, 2020, pp. 677–682.
- [7] M. Gassner, T. Cieslewski, D. Scaramuzza, Dynamic collaboration without communication: vision-based cable-suspended load transport with two quadrotors, in: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 5196–5202.
- [8] N. Michael, J. Fink, V. Kumar, Cooperative manipulation and transportation with aerial robots, *Auton. Rob.* 30 (1) (2011) 73–86.
- [9] Q. Jiang, V. Kumar, The inverse kinematics of cooperative transport with multiple aerial robots, *IEEE Trans. Rob.* 29 (1) (2012) 136–145.
- [10] J.E. Sanz, Quadrotor team modeling and control for DLO transportation, Universidad del País Vasco, 2016 Ph.D. thesis.
- [11] K.K. Dhiman, M. Kothari, A. Abhishek, Autonomous load control and transportation using multiple quadrotors, *J. Aerosp. Inf. Syst.* 17 (8) (2020) 417–435.
- [12] S. Belkhale, R. Li, G. Kahn, R. McAllister, R. Calandra, S. Levine, Model-based meta-reinforcement learning for flight with suspended payloads, *IEEE Rob. Autom. Lett.* 6 (2) (2021) 1471–1478.
- [13] A. Mohiuddin, T. Tarek, Y. Zweiri, D. Gan, A survey of single and multi-UAV aerial manipulation, *Unmanned Syst.* 8 (02) (2020) 119–147.
- [14] M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm robotics: a review from the swarm engineering perspective, *Swarm Intell.* 7 (1) (2013) 1–41.
- [15] F.Y. Hadaegh, S.-J. Chung, H.M. Manohara, On development of 100-gram-class spacecraft for swarm applications, *IEEE Syst. J.* 10 (2) (2014) 673–684.
- [16] K. Saulnier, D. Saldana, A. Prorok, G.J. Pappas, V. Kumar, Resilient flocking for mobile robot teams, *IEEE Rob. Autom. Lett.* 2 (2) (2017) 1039–1046.
- [17] S. Yang, X. Mao, S. Yang, Z. Liu, Towards a hybrid software architecture and multi-agent approach for autonomous robot software, *Int. J. Adv. Rob. Syst.* 14 (4) (2017), doi:[10.1177/1729881417716088](https://doi.org/10.1177/1729881417716088).
- [18] S.-J. Chung, A.A. Paranjape, P. Dames, S. Shen, V. Kumar, A survey on aerial swarm robotics, *IEEE Trans. Rob.* 34 (4) (2018) 837–855.
- [19] K. Baizid, G. Giglio, F. Pierri, M.A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, A. Ollero, Experiments on behavioral coordinated control of an unmanned aerial vehicle manipulator system, in: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 4680–4685.
- [20] A. Din, M. Jabeen, K. Zia, A. Khalid, D.K. Saini, Behavior-based swarm robotic search and rescue using fuzzy controller, *Comput. Electr. Eng.* 70 (2018) 53–65.
- [21] X. Mao, H. Zhang, Y. Wang, Flocking of quad-rotor UAVs with fuzzy control, *ISA Trans.* 74 (2018) 185–193.
- [22] W. Zhao, H. Chu, M. Zhang, T. Sun, L. Guo, Flocking control of fixed-wing UAVs with cooperative obstacle avoidance capability, *IEEE Access* 7 (2019) 17798–17808.
- [23] H. Rezaee, F. Abdollahi, A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots, *IEEE Trans. Ind. Electron.* 61 (1) (2014) 347–354, doi:[10.1109/TIE.2013.2245612](https://doi.org/10.1109/TIE.2013.2245612).
- [24] M. Tognon, C. GABELLIERI, L. Pallottino, A. Franchi, Aerial co-manipulation with cables: the role of internal force for equilibria, stability, and passivity, *IEEE Rob. Autom. Lett.* 3 (3) (2018) 2577–2583, doi:[10.1109/LRA.2018.2803811](https://doi.org/10.1109/LRA.2018.2803811).
- [25] C.W. Reynolds, Flocks, herds, and schools: A distributed behavioral model, in: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*, vol. 21, 1987, pp. 25–34, doi:[10.1145/37401.37406](https://doi.org/10.1145/37401.37406).
- [26] H. Su, X. Wang, Z. Lin, Flocking of multi-agents with a virtual leader, *IEEE Trans. Autom. Control* 54 (2) (2009) 293–307.
- [27] D.K. Villa, A.S. Brandão, M. Sarcinelli-Filho, A survey on load transportation using multirotor UAVs, *J. Intell. Rob. Syst.* (2019) 1–30.
- [28] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajinik, J. Faigl, G. Loianno, V. Kumar, System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization, *Auton. Rob.* 41 (4) (2017) 919–944, doi:[10.1007/s10514-016-9567-z](https://doi.org/10.1007/s10514-016-9567-z).
- [29] V.M. Babu, K. Das, S. Kumar, Designing of self tuning PID controller for AR drone quadrotor, in: *2017 18th International Conference on Advanced Robotics, ICAR 2017*, July, IEEE, 2017, pp. 167–172, doi:[10.1109/ICAR.2017.8023513](https://doi.org/10.1109/ICAR.2017.8023513).
- [30] J. Oyekan, Bio-inspired vision-based leader-follower formation flying in the presence of delays, *Robotics* 5 (3) (2016), doi:[10.3390/robotics5030018](https://doi.org/10.3390/robotics5030018).
- [31] F. Schilling, J. Lecoeur, F. Schiano, D. Floreano, Learning vision-based flight in drone swarms by imitation, *IEEE Rob. Autom. Lett.* 4 (4) (2019) 4523–4530, doi:[10.1109/LRA.2019.2935377](https://doi.org/10.1109/LRA.2019.2935377).
- [32] M. Saska, J. Vakula, L. Preucil, Swarms of micro aerial vehicles stabilized under a visual relative localization, in: *Proceedings - IEEE International Conference on Robotics and Automation, 2014*, pp. 3570–3575, doi:[10.1109/ICRA.2014.6907374](https://doi.org/10.1109/ICRA.2014.6907374).

- [33] C. Virág, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, T. Vicsek, Flocking algorithm for autonomous flying robots, *Bioinspiration Biomimetics* 9 (2) (2014), doi:[10.1088/1748-3182/9/2/025012](https://doi.org/10.1088/1748-3182/9/2/025012).
- [34] K.C. Cao, L. Chunxiang, Formation tracking control and formation stabilization control of multiple nonholonomic mobile robots, in: *Chinese Control Conference, CCC, 2012*, pp. 6053–6058.
- [35] B.C. Nolting, T.M. Hinkelmann, C.E. Brassil, B. Tenhumberg, Composite random search strategies based on non-directional sensory cues, *Ecol. Complexity* 22 (2015) 126–138, doi:[10.1016/j.ecocom.2015.03.002](https://doi.org/10.1016/j.ecocom.2015.03.002).
- [36] J. Oyekan, D. Gu, H. Hu, Visual imaging of invisible hazardous substances using bacterial inspiration, *IEEE Trans. Syst. Man Cybern.* 43 (5) (2013) 1105–1115.
- [37] S. Izumi, S.-I. Azuma, T. Sugie, Multi-robot control inspired by bacterial chemotaxis: coverage and rendezvous via networking of chemotaxis controllers, *IEEE Access* 8 (2020) 124172–124184.
- [38] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, *IEEE Control Syst.* 22 (3) (2002) 52–67, doi:[10.1109/MCS.2002.1004010](https://doi.org/10.1109/MCS.2002.1004010).
- [39] J. Oyekan, H. Hu, Ant robotic swarm for visualizing invisible hazardous substances, *Robotics* 2 (1) (2013) 1–18.
- [40] D.A. Brown, H.C. Berg, Temporal stimulation of chemotaxis in *Escherichia coli*, *Proc. Natl. Acad. Sci. U. S. A.* 71 (4) (1974) 1388–1392, doi:[10.1073/pnas.71.4.1388](https://doi.org/10.1073/pnas.71.4.1388).
- [41] J. Oyekan, D. Gu, H. Hu, A model for using self-organized agents to visually map environmental profiles, *Ecol. Complexity* 19 (2014) 68–79, doi:[10.1016/j.ecocom.2014.04.004](https://doi.org/10.1016/j.ecocom.2014.04.004).
- [42] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in: *Proceedings - IEEE International Conference on Robotics and Automation*, 1985, pp. 500–505, doi:[10.1109/ROBOT.1985.1087247](https://doi.org/10.1109/ROBOT.1985.1087247).
- [43] A. Zakarian, J.W. Knight, L. Baghdasaryan, Modelling and analysis of system robustness, *J. Eng. Des.* 18 (3) (2007) 243–263, doi:[10.1080/09544820600804939](https://doi.org/10.1080/09544820600804939).
- [44] J. Fink, N. Michael, S. Kim, V. Kumar, Planning and control for cooperative manipulation and transportation with aerial robots, *Int. J. Rob. Res.* 30 (3) (2011) 324–334, doi:[10.1177/0278364910382803](https://doi.org/10.1177/0278364910382803).
- [45] N. Michael, J. Fink, V. Kumar, Cooperative manipulation and transportation with aerial robots, *Auton. Rob.* 30 (1) (2011) 73–86, doi:[10.1007/s10514-010-9205-0](https://doi.org/10.1007/s10514-010-9205-0).