



Intelligent obstacle avoidance algorithm for safe urban monitoring with autonomous mobile drones

Didar Yedilkhan^a, Abzal E. Kyzyrkanov^{a,b,*}, Zarina A. Kutpanova^{a,c},
Shadi Aljawarneh^d, Sabyrzhan K. Atanov^b

^aDepartment of Computer Engineering, Astana IT University, Astana, 010000, Kazakhstan

^bDepartment of Computer and Software Engineering, L. N. Gumilyov Eurasian National University, Astana, 010000, Kazakhstan

^cDepartment of Automation and Control Systems, L. N. Gumilyov Eurasian National University, Astana, 010000, Kazakhstan

^dDepartment of Software Engineering, Jordan University of Science and Technology, Irbid, 22110, Jordan

ARTICLE INFO

Publishing editor: Yu-Lian He

Keywords:

Drone swarms

Fuzzy logic

Intelligent solution

Smart city

Urban monitoring

ABSTRACT

The growing field of urban monitoring has increasingly recognized the potential of utilizing autonomous technologies, particularly in drone swarms. The deployment of intelligent drone swarms offers promising solutions for enhancing the efficiency and scope of urban condition assessments. In this context, this paper introduces an innovative algorithm designed to navigate a swarm of drones through urban landscapes for monitoring tasks. The primary challenge addressed by the algorithm is coordinating drone movements from one location to another while circumventing obstacles, such as buildings. The algorithm incorporates three key components to optimize the obstacle detection, navigation, and energy efficiency within a drone swarm. Firstly, the algorithm utilizes a method for calculating the position of a virtual leader, acting as a navigational beacon to influence the overall direction of the swarm. Secondly, the algorithm identifies observers within the swarm based on the current orientation. To further refine obstacle avoidance, the third component involves the calculation of angular velocity using fuzzy logic. This approach considers the proximity of detected obstacles through operational rangefinders and the target's location, allowing for a nuanced and adaptable computation of angular velocity. The integration of fuzzy logic enables the drone swarm to adapt to diverse urban conditions dynamically, ensuring practical obstacle avoidance. The proposed algorithm demonstrates enhanced performance in the obstacle detection and navigation accuracy through comprehensive simulations. The results suggest that the intelligent obstacle avoidance algorithm holds promise for the safe and efficient deployment of autonomous mobile drones in urban monitoring applications.

1. Introduction

Urban monitoring has become a critical concern in contemporary times due to the escalating challenges posed by rapid urbanization based on the statistics provided by the United Nations. As cities expand and populations grow, managing resources effectively, enhancing public safety, and mitigating the environmental impact become imperative. Monitoring urban environments provides invaluable insight into traffic patterns, air quality, and other crucial parameters, enabling the development of more intelligent,

* Corresponding author. Department of Computer Engineering, Astana IT University, Astana, 010000, Kazakhstan.

E-mail addresses: d.yedilkhan@astanait.edu.kz (D. Yedilkhan), abzall@gmail.com (A.E. Kyzyrkanov), z.kutpanova@astanait.edu.kz (Z.A. Kutpanova), saaljawarneh@just.edu.jo (S. Aljawarneh), atanov5@mail.ru (S.K. Atanov).

<https://doi.org/10.1016/j.jnlest.2024.100277>

Received 12 May 2024; Received in revised form 31 July 2024; Accepted 14 August 2024

Available online 22 August 2024

1674-862X/© 2024 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

sustainable cities [1]. This emphasis on urban monitoring is pivotal for informed decision-making, resource optimization, and creating urban space that is resilient, efficient, and conducive to high-quality life.

Most urban monitoring solutions and concepts rely on the Internet of things [2], wherein a network of sensors is strategically positioned throughout the urban area to gather relevant data. Usually, these sensors are positioned in predetermined spots, leading to biases in data collection and the risk of blind spots in case of sensor malfunctions. An alternative approach entails placing sensors on mobile platforms, such as regular taxis or public transportation. Placing sensors on moving transport enhances urban monitoring by providing real-time data on the vehicle location and various conditions, facilitating efficient urban planning and resource allocation.

The ongoing advancement of robotics and autonomous systems plays a crucial role in influencing the achievement of urban monitoring. One notable category within this domain is consumer unmanned aerial vehicles (UAVs) or drones, denoting aircraft without an onboard crew. Drones exhibit varying levels of autonomy, ranging from remote control to full automation, and differ in the design, purpose, and various other characteristics. Drones have emerged as the swiftest-growing autonomous systems of the past decade, finding increasingly prevalent applications, particularly in urban monitoring.

It is well-established that sensor-equipped drones can access areas which are inaccessible to human presence, facilitating information acquisition through large-scale remote sensing images. They can capture detailed urban data encompassing city traffic, pedestrians, and air quality concentrations [3]. Operationally, drones are characterized by ease of use and support functions such as automatic detection and cruising, thereby significantly streamlining operational tasks. In other words, drones are continually developing, giving rise to various types of autonomous systems tailored for diverse tasks.

Incorporating UAVs alongside mobile sensors expands urban monitoring capabilities by offering aerial perspectives and covering areas inaccessible to ground-based sensors [4]. UAVs can swiftly survey large areas, providing high-resolution imagery and data for various urban monitoring applications, such as traffic management, disaster response, and infrastructure inspection, complementing the data gathered by mobile sensors to offer comprehensive understanding of urban dynamics and challenges.

Several literature reviews have explored the applications and challenges associated with drones in the context of smart cities in general and urban monitoring in particular. In Ref. [5], Gohari et al. provided a systematic review based on search and meta-analysis in the Web of Science and Scopus research databases. They grouped the applications of drones in smart cities based on categories such as Transportation, Environment, Infrastructure, Object or People Detection, Disaster Management, and General Data Collection. As they mentioned in the research, most papers focused on traffic monitoring, traffic safety, and parking management.

For traffic monitoring, models and methods using drones in the simulated environment have been proposed in Refs. [6–8], which also answered the question of how drones can be used for traffic monitoring and management. Reference [9] reports a model to obtain complete coverage of the city roads using drones and lightweight semantic neural networks. All these studies employed simulation methodologies to investigate and analyze various facets of drone applications.

Certain scholars restricted their investigation to specific zones within the urban landscape, narrowing their focuses from the entirety of the city to the designated parking areas. In Ref. [10], Gogoi et al. endeavored to enhance parking monitoring and management precision by deploying drones characterized by low power consumption. Similar to the preceding instances, simulation methodologies were also employed in these cases.

A noteworthy methodology has been introduced in Ref. [11], wherein Shirazi et al. advocated for a system integrating drones with video image recognition technologies. The proposed system demonstrated efficacy in tracking vehicles and offering superior traffic measurements compared to conventional fixed closed-circuit television cameras positioned at intersections. They implemented the computer vision technology to underscore the enhanced accuracy achieved by drones in this context. Another application of drones in traffic monitoring and management has been presented in Ref. [12]. Beg et al. proposed an intelligent and autonomous solution using drones to detect real-time emergency traffic situations and investigate potential accidents with rapid response units.

The preceding studies predominantly illustrate the utilization of drones in transportation; however, a substantial body of literature details the deployment of drones for air pollution monitoring. Experiments have been conducted to devise an air quality measurement system reliant on unmanned aerial vehicle-ground control station (UAV-GCS) communications, as documented in Ref. [4]. Additionally, several research groups have successfully endeavored to design and integrate sensors for the measurement of particulate matter in specific urban areas, utilizing autonomous drones [13–16]. Gao et al. proposed a system based on 360-degree aerial images with error and energy use for air quality monitoring with high accuracy [17].

Naturally, drones have a significant practical application in transportation and air quality monitoring. Therefore, their applications are extending to various related domains, including bridge inspection [18], power line inspection [19,20], pavement inspection [21,22], and building inspection [23].

Using multiple drones, as opposed to a single drone, offers several advantages in various applications. One of the key benefits is the increased coverage and efficiency. Multiple drones can collectively cover larger areas in less time, providing comprehensive monitoring and data collection. This is particularly advantageous in scenarios where extensive spatial coverage is crucial, such as surveillance in large urban areas, agricultural fields, or disaster-stricken regions. Moreover, the deployment of multiple drones enhances redundancy and reliability. In the event of technical failure or unforeseen challenges, other drones can compensate for the issues, ensuring the continuous operation and data acquisition. This redundancy is vital for maintaining the integrity of the mission and obtaining accurate and reliable results [24]. The emergence of swarming UAV systems or drone swarms represents an innovative approach that amalgamates the coordinated efforts of multiple unmanned drones to execute complex missions more effectively as a collective entity [25]. Using a fleet of drones enables the implementation of distributed sensing strategies. Each drone can be equipped with specific sensors tailored to different aspects of the task, leading to a more specialized and efficient data collection process. This approach enhances the overall capabilities of the drone fleet, making it adaptable to a wide range of applications.

Collaboration among multiple drones also allows for executing complex tasks and missions that may be challenging or impossible for

a single drone to accomplish. For instance, in tasks, such as search and rescue, environmental monitoring, or infrastructure inspection, multiple drones can collaborate to cover different perspectives, angles, or heights, providing more comprehensive understanding of the situation. However, solving the tasks of managing a swarm of drones and organizing interactions between “individuals” is generally a significant difficulty.

The effectiveness of using drone swarms largely depends on the chosen management strategy. There are centralized and decentralized strategies. The central device can be located both outside the group (such as, on the operator’s control panel) and on one of the group members (centralized control “with a master”). Centralized strategies perform well when managing small groups of robots. The load on the communications channel and control devices increases as the number increases. One solution is to implement a hierarchical model in which the group is divided into subgroups, each controlled by its leader.

This study makes significant contributions to the field of urban monitoring by introducing an innovative intelligent obstacle avoidance algorithm specifically designed for coordinating drone swarms in urban environments. The novelty of this research lies in the integration of a virtual leader-follower approach with fuzzy logic to dynamically adjust drone velocities, enabling efficient navigation and obstacle avoidance. By optimizing energy consumption and improving the obstacle detection accuracy, this algorithm enhances the overall performance and reliability of drone swarms in complex urban landscapes. These advancements contribute to the development of smarter, more resilient urban monitoring systems, ultimately facilitating better resource management, enhanced public safety, and reduced environmental impact in rapidly growing cities.

2. Problem statement

The algorithm presented in this paper seeks to address a suite of intricate challenges inherent in the coordinated movement of a drone swarm, particularly within the safe urban monitoring context. These challenges arise from the need to ensure efficient, synchronized navigation of multiple drones while maintaining a specific formation and minimizing energy consumption. This section will introduce the core problems that the algorithm aims to solve.

Firstly, coordinating drone movements in a swarm while preserving a specific formation is a significant challenge. The algorithm employs a leader-follower approach with a virtual leader to address this issue. This method allows for effective drone coordination, ensuring that the entire swarm moves harmoniously as a single entity. Secondly, energy efficiency is paramount in prolonged monitoring tasks. The algorithm introduces a strategy to reduce energy consumption by selectively turning off the drones’ rangefinders. This approach enables the drones to avoid obstacles without requiring all drones to have their rangefinders continually active, thereby conserving energy. Lastly, the algorithm incorporates fuzzy logic for calculating the velocities of the drones. This inclusion is crucial for dealing with the uncertainty and variability inherent in real-world urban environments.

To tackle these challenges, the current research considers two-dimensional (2D) formations to simplify the initial development and testing of the intelligent obstacle avoidance algorithm. This approach provides valuable insight into the algorithm’s capabilities and performance in managing drone swarms in urban environments. It is also sufficient for most cases, as drones typically fly with 2D formation patterns and can consider obstacles as 2D objects. A notable strength of this study is that the drones make decisions dynamically based on the real-time lidar detections without prior knowledge of obstacles, underscoring the algorithm’s adaptability and effectiveness in a dynamic and changing environment. Extending the algorithm to three-dimensional (3D) space and handling moving obstacles will be explored in future work.

2.1. Leader-follower approach

In swarm robotics, the leader-follower approach plays a crucial role in managing the coordinated movement of multiple autonomous agents. In this method, followers are programmed to maintain a consistent distance from a designated leader. The relationship between the leader and each follower can be mathematically represented as

$$\sqrt{(x_i - x_L)^2 + (y_i - y_L)^2} = d_i \quad (1)$$

where (x_L, y_L) represents the position of the leader, (x_i, y_i) represents the position of the i th follower drone, and d_i is the constant distance that the i th follower drone tries to maintain from the leader (Fig. 1).

While the traditional leader-follower approach provides structured formation maintenance, it poses a significant risk when a single physical drone is designated as the leader. If this leader drone encounters a malfunction or crash, the coordinated movement of the entire swarm can be disrupted, leading to system failure.

The algorithm proposed in this paper employs an innovative adaptation to mitigate this risk. Instead of assigning the leadership role to a physical drone, the algorithm designates the center of the geometric pattern that the swarm maintains as the virtual leader. This virtual leader is not a physical entity but a dynamically calculated point representing the formation’s ideal central position. By doing so, every drone in the swarm attempts to maintain a specific distance from this virtual leader position. This approach preserves the formation integrity and enhances the system’s robustness. With no single drone being pivotal to the leadership role, the risk of system-wide failure due to the loss of one drone is significantly mitigated. The swarm, therefore, exhibits greater resilience and adaptability, as well as qualities that are indispensable in complex and unpredictable urban monitoring scenarios.

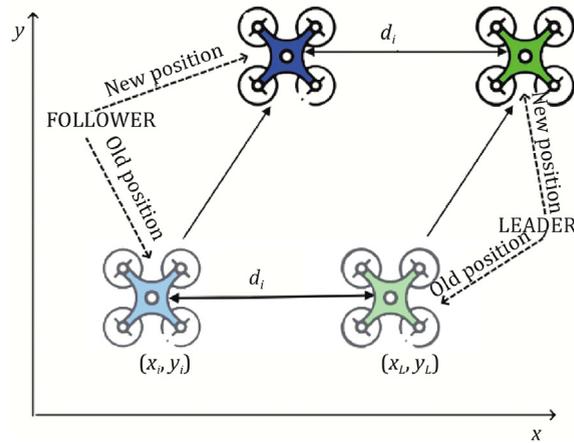


Fig. 1. Leader-follower approach.

2.2. Optimizing energy consumption in swarm robotics

One of the predominant challenges in swarm robotics, especially for prolonged missions, is energy consumption. As these systems are often deployed extensively, efficient energy usage becomes paramount. The usage of rangefinders, crucial for the obstacle detection and navigation, epitomizes this challenge. Contemporary rangefinders are technologically advanced, capable of identifying obstacles from considerable distances, sometimes spanning several times the overall size of the swarm. However, this impressive functionality comes at a cost. These devices require huge energy consumption, and their continuous operation can substantially drain the power reserves of the drones, limiting the duration and scope of missions.

To mitigate the issue of energy depletion, the algorithm presented in this paper proposes a solution centered on the strategic activation of rangefinders. The core principle of this approach is based on the fact that not all drones in the swarm need always to have their rangefinders operational. The algorithm posits that drones positioned in the rear of the swarm can rely on the movement and navigational decisions of the drones at the front. Similarly, those in the middle can navigate effectively by following cues from drones positioned on their left and right sides.

The algorithm suggests a selective activation strategy for rangefinders in line with this principle. Specifically, it proposes to activate the rangefinders of only three drones: The drone at the forefront of the swarm and the drones on the extreme left and right sides. This targeted activation ensures that essential navigational information is acquired while significantly reducing overall energy consumption.

The potential impact of this optimization strategy is multifold. Firstly, it directly contributes to extending mission duration by conserving energy. With fewer rangefinders active at any given moment, the swarm's power consumption decreases, allowing for more extended operational periods. Secondly, this approach enhances the operational resilience of the swarm. By reducing dependency on the continuous functioning of every rangefinder, the system becomes less susceptible to failure of single component. Lastly, it contributes to the scalability of swarm operation. As energy efficiency improves, deploying more enormous swarms or extending the operational range becomes more feasible.

2.3. Application of fuzzy logic in velocity calculation

Fuzzy logic, originating in fuzzy set theory, offers a compelling solution to handling uncertainty and imprecise data in swarm robotics. This form of many-valued logic diverges from traditional binary logic by acknowledging and working with degrees of truth rather than absolutes. Such an approach is particularly advantageous in the dynamic and unpredictable scenarios the swarm robots usually encounter.

The selection of fuzzy logic for velocity calculation is rooted in its intrinsic capability to process ambiguous and incomplete information. In the environment where swarm robots operate, sensor readings and environmental variables are seldom clear-cut or error-free. Fuzzy logic allows for decision-making based on these partial truths, which is crucial in such settings.

In the context of this paper, the application of fuzzy logic is pivotal in calculating velocities, assisting the drones in the swarm to navigate efficiently and effectively amidst ambiguous or incomplete information. The intricacy of this application, including fuzzification, the establishment of fuzzy rules, and defuzzification, will be further explored in the upcoming sections, elucidating the integral role of fuzzy logic in augmenting the capabilities of the swarm.

3. Methodology

The methodology section describes the comprehensive processes and techniques employed in developing and evaluating the algorithm that orchestrates the movement of the swarm. It provides an in-depth exposition of the algorithm's design, including its underlying principles. The focus is on the algorithm's ability to facilitate coordinated, obstacle-avoidant navigation while preserving a

predefined geometric pattern among the drones. It also aims to highlight the algorithm's applicability and potential impact in urban monitoring, paving the way for more efficient and sophisticated approaches.

The algorithm introduced in this paper orchestrates a coherent and strategic movement of a swarm of drones towards a predefined destination. This process is activated as the swarm sets off, with each drone updating its velocity vectors regularly. Central to this algorithm is the concept of a virtual leader, whose position is dynamically calculated during each interval, guiding the trajectory of the swarm.

Each drone's motion is characterized by a velocity vector, denoted as a pair (V, W) . Here, V represents the linear velocity, quantified in meters per second (m/s), and W indicates the angular velocity or the rotation angle. While the linear velocity remains consistent throughout the journey, it is the direction or angular velocity, which is continually adjusted to maneuver through the environment adeptly.

The algorithm unfolds through several pivotal steps:

- 1) Virtual leader's position calculation: The initial phase involves using formulas to pinpoint the position of the virtual leader. This virtual leader acts as a navigational beacon, influencing the general direction of the swarm.
- 2) Identification of robot observers: Based on the current orientation of the swarm, certain drones equipped with active rangefinders are designated as observers. These observers are integral for navigation and circumventing obstacles. To enhance energy efficiency, all other drones in the swarm deactivate their rangefinders. In most scenarios, three observer drones are adequate: One centrally positioned in the leading formation (the middle observer) and two flank drones (the left and right observers). When the swarm necessitates a directional shift, the rangefinders on the left and right observers are subtly angled to optimize the obstacle detection.
- 3) Angular velocity calculation via fuzzy logic: The angular velocity is ascertained using fuzzy logic. This method assesses the proximity to any detected obstacles via the operational rangefinders and the target's location. The integration of fuzzy logic facilitates a more nuanced and adaptable computation of angular velocity, enabling the swarm to dynamically adapt to diverse environmental conditions.

The drones within the swarm periodically update their angular velocities by executing these steps, ensuring an efficient and synchronized advancement toward the target. If some drones do not keep up with the team due to disconnection or crashes, the algorithm can handle this by simply ignoring the unavailable drones. Each drone continuously updates its angular velocity, identifies its observers, and calculates the virtual leader's position based on data retrieved from the remaining connected drones. Consequently, the formation pattern is maintained despite the absence of some drones. The ensuing subsections of this paper will delve deeper into each algorithmic step, providing an in-depth exploration of its mechanisms and practical applications.

3.1. Calculation of the position of the virtual leader

In the sophisticated action of the coordinated movement that defines the swarm's behavior, the concept of a 'virtual leader' plays a central role. This virtual leader is not a physical entity but an algorithmically determined point guiding the swarm's collective movement. Calculating this virtual leader's position is pivotal in ensuring the swarm's cohesive and patterned navigation.

The position of the virtual leader is determined as the average of what could be termed the "ideal virtual leader's position" for each robot within the swarm. This ideal position is a theoretical point that would exist if each individual robot was perfectly positioned within the desired formation. It represents where the virtual leader would be for each drone if everything was ideally aligned.

The mathematical representation for the position of the virtual leader (x_L, y_L) is given by

$$x_L = \frac{\sum_i x_i - x_{i,form}}{N} \quad (2a)$$

$$y_L = \frac{\sum_i y_i - y_{i,form}}{N} \quad (2b)$$

where (x_L, y_L) represents the coordinate of the virtual leader and (x_i, y_i) denotes the current position of the i th robot, while $(x_{i,form}, y_{i,form})$ indicates the desired position of the i th robot relative to the leader—essentially where it should be to maintain the formation. The variable N stands for the number of active (non-collided) robots in the swarm at the given time step.

An important consideration is the fact that the virtual leader is programmatically positioned at the center of the desired geometric pattern maintained by the swarm. By calculating the virtual leader's position as the mean average of the positions of all robots, while adjusting for the desired formation by subtracting $x_{i,form}$ and $y_{i,form}$, the algorithm ensures the maintenance of the geometric pattern. Here, the relation between (x_i, y_i) and $(x_{i,form}, y_{i,form})$ is that $(x_i - x_{i,form}, y_i - y_{i,form})$ gives the position of the virtual leader if the i th drone is ideally maintaining the formation. This adjustment becomes particularly crucial in scenarios where some robots may have collided or crashed or be unavailable, thereby potentially disrupting the formation.

This calculated position of the virtual leader becomes a guiding beacon, influencing the trajectory of each robot and, thereby, the entire swarm. It ensures that even in the face of individual discrepancies or unexpected events, the swarm can continue to move as a cohesive and coordinated entity, adhering to the intended geometric pattern.

3.2. Identification of robots—observers

Identifying observer agents with active rangefinders is integral to optimizing the energy efficiency of the swarm. These observers are selected based on the swarm’s current direction of motion. Only three observers are activated to conserve energy: The leading robot in the center (central observer) and two robots positioned on the left and right sides. The other robots in the swarm deactivate their rangefinders.

For the selection of these observers, a coordinate transformation is applied. This involves shifting the origin to the virtual leader’s position (x_L, y_L) and rotating the x -axis to align with the swarm’s direction of motion α (Fig. 2). The new coordinate (x'_i, y'_i) of the i th robot located in (x_i, y_i) can be calculated by

$$x'_i = (x_i - x_L)\cos \alpha + (y_i - y_L)\sin \alpha \tag{3a}$$

$$y'_i = -(x_i - x_L)\sin \alpha + (y_i - y_L)\cos \alpha. \tag{3b}$$

The process for identifying observers is as follows:

- 1) Central observer selection: In the transformed coordinate system, the robot positioned ahead of all others in the direction of motion is selected as the central observer. This is the robot with the maximum x'_i value. If multiple robots share this maximum value, the one closest to the x -axis (i.e. with the lowest absolute value of y'_i) is chosen. For instance, as depicted in Fig. 3 (a), if Robots 3 and 4 share the maximum x'_i value, Robot 3 is chosen as it is closer to the x -axis.
- 2) Left and right observers selection: The left and right observers are identified based on their positions relative to the central axis. The left observer is the robot with the highest y'_i value, while the right observer has the lowest. If multiple robots meet the criteria, the one with the highest x'_i value is selected. In Fig. 3 (a), Robot 2 is chosen over Robot 1 as the left observer because of its greater x'_i value, despite both having the same maximum y'_i value.
- 3) Special cases handling: When one robot qualifies as both a central observer and a side observer, it is assigned as the side observer. The next robot that meets the criteria is chosen as the central observer. This situation is shown in Fig. 3 (b), where Robot 5 meets the criteria for both the central and right observers but is chosen as the right observer. Robot 4, which is next in line, is chosen as the central observer.
- 4) Observers in reduced numbers: If only one drone is available (with all others being unavailable or crashed), it operates as the central observer. If there are two drones, they operate as the left and right observers.

By following these steps, the swarm efficiently selects the observer robots. These observers, with their active rangefinders, play a crucial role in determining the swarm’s movement direction—straight, left, or right—based on detected obstacles and environmental conditions.

3.3. Calculation of velocity using fuzzy logic

In autonomous robotic navigation, the modulation of angular velocity plays a pivotal role in achieving precise and fluid movements. This subsection introduces fuzzy logic for calculating the angular velocity while maintaining a constant linear velocity, denoted as V . The constancy of V ensures a stable forward progression, with its value predetermined before movement initiation. Integral to our methodology is the secondary configurable parameter, the maximum angular velocity adjustment, w . This parameter delineates the bounds within which the angular velocity may be variably adjusted and is integral to the membership functions for both the input and

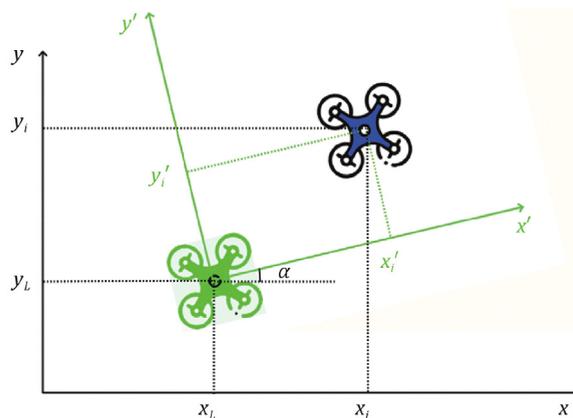


Fig. 2. Coordinate transformation.

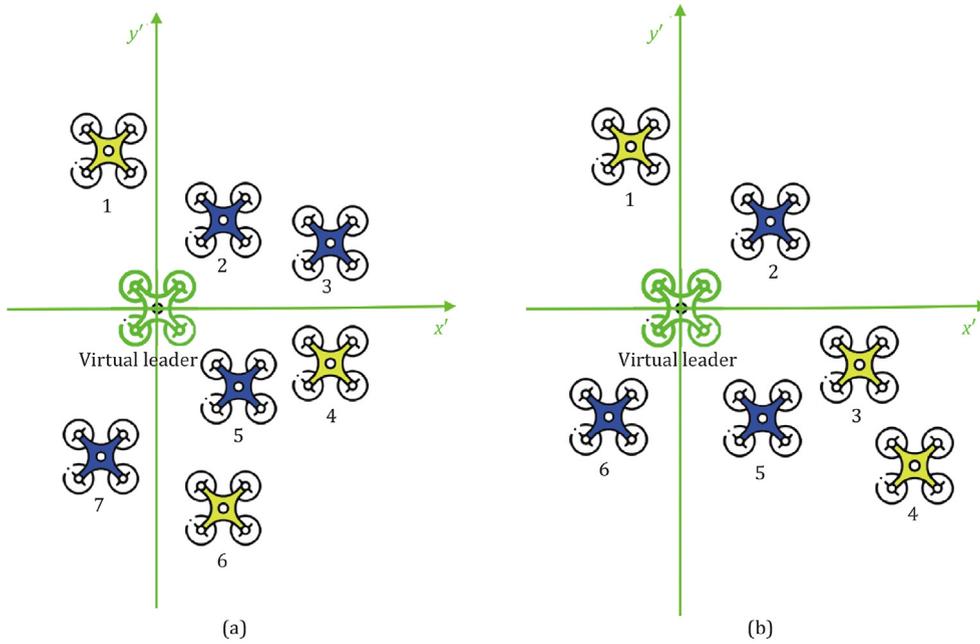


Fig. 3. Selection of observers in (a) normal case and (b) case when one robot qualifies as both a central observer and a side observer.

output variables within our fuzzy logic system.

The algorithm proposed operates on a quintet of inputs: The extant angular velocity, the intended directional goal, and the proximal assessments from a triad of rangefinders. A singular output is derived from these inputs: The corrective angular step, α . This calculated step is then applied to adjust the current angular velocity (W_{current}), yielding a new angular velocity, W_{new} , articulated by

$$W_{\text{new}} = W_{\text{current}} + \alpha \tag{4}$$

where α is the value extracted through a defuzzification process. The value of α bounded within the range $(-w, w)$, where w is the adjustable parameter that defines the maximum angular velocity adjustment. This ensures that the adjustment remains within the predefined limits of system capabilities. Subsequent subsections will delve into the intricacy of the fuzzy logic implementation, detailing the membership functions, rule base, and defuzzification strategy employed to achieve responsive and adaptive angular velocity control.

3.3.1. Fuzzification and membership function

Fuzzification is the foundational process in fuzzy logic, where crisp input values are transformed into fuzzy values based on predefined linguistic terms. These linguistic terms, often called literals, represent categories or labels in the fuzzy set theory. Membership functions play a pivotal role in characterizing these literals, assigning each a degree of membership between 0 and 1. In this context, a value of 0 signifies no membership, while a value of 1 indicates full membership to the particular literal.

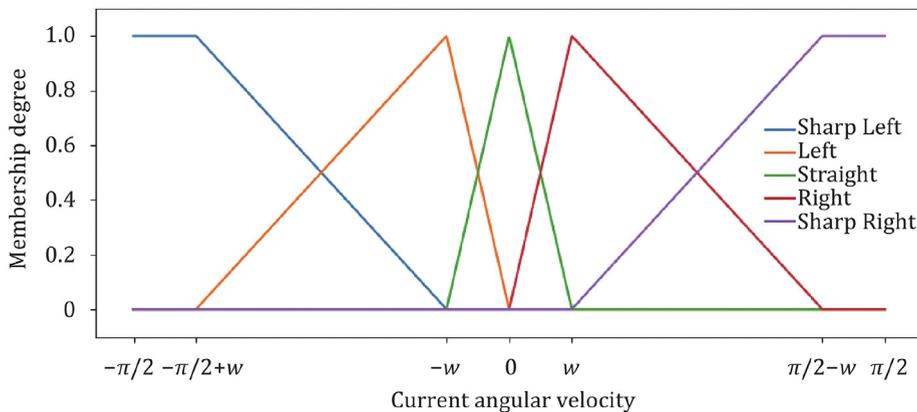


Fig. 4. Membership functions for current angular velocity.

As we venture further in this subsection, we will provide detailed information about each input, elucidating their associated literals and the mathematical equations that define their membership functions.

The “current angular velocity” is an essential input variable to our fuzzy logic system. We have defined this input through five linguistic literals: Sharp Left, Left, Straight, Right, and Sharp Right. These literals classify the current angular velocity into discernible behavioral attributes. A triangular membership function defines each literal, as visualized in Fig. 4.

The literal Sharp Left represents a significant anti-clockwise angular velocity. The peak of this function, where the membership is the highest, is at $x = -\pi/2$. This describes a situation where the robot’s movement is the most sharply towards the left. The Left literal suggests a milder anti-clockwise angular velocity than Sharp Left. The peak of this function is found at $x = -w$, representing a moderate leftward movement. The Straight literal is indicative of no angular velocity, implying a forward movement. The peak, signifying the highest membership, is located at $x = 0$. The Right literal stands for a mild clockwise angular velocity. The function’s peak, showing the maximum membership, is located at $x = w$, representing a gentle rightward movement. Lastly, the Sharp Right literal denotes a substantial clockwise angular velocity. The highest membership of this function occurs at $x = \pi/2$, representing a pronounced rightward direction of movement.

The second vital input to our fuzzy logic system is the “goal side”. It serves as a determinant for the target position relative to the robot’s current movement direction and its virtual leader. Specifically, this input aids the robot in navigating towards the goal side when no obstacles are detected, or when facing ambiguity in the situational assessment from both left and right observers. Under such conditions, the robot would prioritize moving in the direction of the target. The corresponding membership functions for this input are detailed in Fig. 5.

The goal side has been characterized using three linguistic literals: Left, Straight, and Right. Each of these literals helps discern the direction of the target side, and the triangular membership functions define them. The Left literal indicates that the goal side is positioned to the left relative to the robot’s current direction of movement. The Straight literal portrays a scenario where the goal side is directly ahead, in alignment with the robot’s current direction. Lastly, the Right literal signifies that the goal is positioned to the robot’s right about its current movement direction.

The “rangefinder detections” of the left, front, and right observers are crucial for our robotic navigation system. These detections help the robot assess its immediate surroundings, allowing it to adapt its movement pattern in real-time to circumvent potential obstacles and hazards. The variable sensor_range within these functions is a pivotal parameter, representing the maximum distance the rangefinder can measure. This range sets the boundary within which objects can be detected. For each observer, there are three linguistic literals: Far, Close, and Not Detected. The membership functions of these literals, except the literal Not Detected, are presented in Fig. 6. Far and Close are represented by continuous membership functions, whereas Not Detected is a binary state that cannot be visualized on a continuous spectrum and hence is not depicted in Fig. 6. Each literal aids in interpreting the proximity of detected objects or obstacles relative to the robot.

The Close literal signifies that an object is nearby, potentially necessitating the robot to alter its course to prevent a collision. The Far literal implies that the detected object is substantially from the robot, posing no immediate hindrance to its current path.

The Not Detected literal is distinct in its functionality. Instead of indicating a position or distance, it represents the absence of the object detection. If the rangefinder does not detect any obstacles, this literal returns true, while in the presence of the detection, it returns false.

3.3.2. Fuzzy output and defuzzification

Fuzzy output is the direct result of a fuzzy inference system where input variables are processed through fuzzy rules. In the context of controlling angular velocity, the fuzzy output represents the approximate adjustments to be made to the current angular velocity. In our context, the fuzzy output is an approximation in angular velocity, denoted as α , which adjusts the current angular velocity W_{current} to a new value $W_{\text{current}} + \alpha$, within the range $[-w, w]$. This adjustment, denoted as α , is determined within a range of $-w$ to w , where positive values indicate the rightward turn and negative values represent the leftward turn. The fuzzy output consists of four literals:

- 1) Left: This suggests a decrease in the angular velocity, moving towards $-w$.
- 2) Straight: It implies maintaining the current angular velocity, suggesting no change.

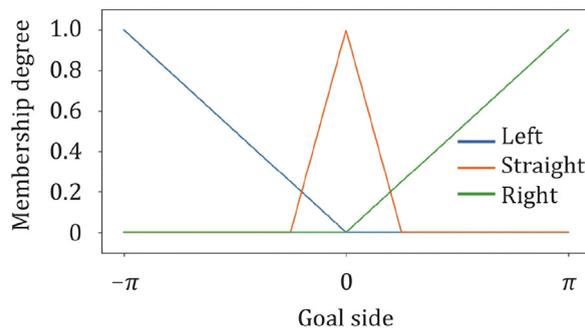


Fig. 5. Membership functions for current angular velocity.

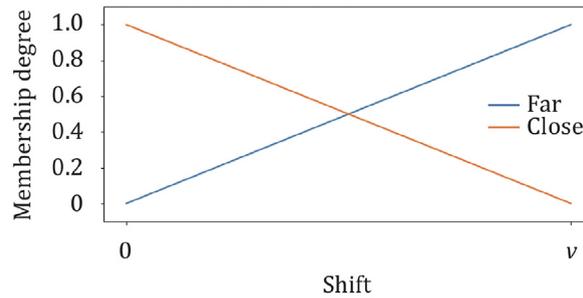


Fig. 6. Far and Close membership functions.

- 3) Right: This indicates an increase in the angular velocity, up to w .
- 4) Random: This is selected when a change is necessary, but there is no clear preference for left or right. It helps avoid potential deadlocks that could occur from consistently choosing the same direction in a state of equilibrium.

Centroid defuzzification is employed to transform the fuzzy output into a crisp value. This method calculates the center of gravity of the output fuzzy set, providing a single output value from the overlapped results of fuzzy rules. The choice of centroid defuzzification ensures a balanced decision-making process that considers the contribution of all applicable rules and their respective membership degrees, leading to a decision representing the consensus of the fuzzy logic system.

It is important to note that while the Random literal may initially seem counterintuitive, it plays a crucial role in dynamic environments. The random choice between Left and Right when conditions are equal can prevent the system from stalling and provide an exploration element essential for navigating through complex scenarios. However, this approach is used sparingly, as deterministic approaches using Left or Right usually guide the system more efficiently. Randomness introduces variability, which can prevent the system from getting stuck in repetitive patterns that do not lead to a resolution, but using it excessively can lead to unpredictable behaviors.

3.3.3. Fuzzy rules

Fuzzy rules are the core of the fuzzy logic system, functioning as a series of if-then statements that utilize linguistic variables to define the system's behavior. These rules are crucial as they encapsulate the expert knowledge and decision-making process in a format that the fuzzy logic system can process, allowing it to make inferences under uncertainty or imprecision. These inferences, drawn from fuzzy rules, enable the system to react with nuanced responses similar to human reasoning, making fuzzy logic particularly effective in complex, real-world scenarios where binary logic falls short.

The subsequent discourse delves into the intricacy of fuzzy rules and their application in the fuzzy logic system. These rules form the cornerstone of the fuzzy inference process, acting as the conduit through which fuzzified inputs are transformed into actionable outputs. The efficacy of a fuzzy logic system hinges on its rule base—a well-conceived compilation of if-then statements that encapsulate the system's expert knowledge and strategic decision-making. The subsections that follow will progressively unfold the individual rules composing the system's rule base, illuminating their construction, interactions, and the intricate reasoning they represent.

3.3.3.1. Rule for no obstacle detection. In scenarios where no obstacles are detected by the rangefinders, the decision for the vehicle's direction is largely dependent on the goal side. This simplification streamlines the process and ensures that the vehicle's path is goal-oriented.

When the drone is moving Straight with respect to its current angular velocity, the goal side's input takes precedence in dictating the output. Should the goal side be Left, the system outputs a Left directive to adjust the angular velocity accordingly. Similarly, a Right goal side results in a Right output. If the goal side is directly ahead (Straight), the output remains Straight, indicating no change in the angular velocity.

If the vehicle's current angular velocity is Left, the system considers the direction to the goal side to determine the appropriate output. If the goal side is also to the Left, the system outputs Straight, suggesting that no change is needed—the vehicle is already on course. Conversely, if the goal side is either Straight or to the Right, the output switches to Right, decreasing the angular velocity's absolute value and gently steering the vehicle back towards the goal.

The case where the vehicle's current angular velocity is Right follows the inverse logic. Should the goal side be Right, the system maintains the direction with a Straight output. If the goal side is Straight or to the Left, the output becomes Left, reducing the absolute angular velocity to the Left, aligning the vehicle's trajectory with the intended target. This set of rules ensures that the vehicle's path is corrected smoothly and remains aligned with the target direction while preventing unnecessary deviations.

3.3.3.2. Rule for obstacle detection (general case). In scenarios where an obstacle is detected, the complexity of the fuzzy rule set increases due to the need to evaluate the proximity of obstacles from multiple observers.

If a robot's current angular velocity is oriented Straight ahead and encounters obstacles, a predefined fuzzy logic rule set guides the adjustment of the robot's trajectory. The decision-making process for the adjustment prioritizes the detections from the left and right

observers as follows:

- a) Rule for Straight angular velocity with unequal obstacle detections. If the detection from the right observer indicates an obstacle is closer compared with the left observer (Fig. 7 (a)), then the output action will be to turn to the Left. Conversely, if the left observer detects an obstacle that is closer than what the right observer detects (Fig. 7 (b)), the robot will turn to the Right.
- b) Rule for Straight angular velocity with equal obstacle detections. In the instance where both the left and right observers detect obstacles at equal proximity (Fig. 8), the decision will be influenced by the goal side. The output will be Right if the goal side is to the Right; it will be Left if the goal side is to the Left. If the goal side is directly ahead (front), the output will default to Random. This randomness is introduced to prevent the robot from proceeding straight forward, which would risk an obstacle collision, yet allows for a decision when the left and right detections are balanced.

By applying these rules, a robot can dynamically adjust its path in real-time, effectively responding to the immediate environmental conditions to maintain a collision-free trajectory towards the designated goal.

3.3.3.3. Rules for obstacle detection (when veering to the Left/Right). In scenarios where a robot is already veering to the Left and encounters obstacles, the focus shifts from the goal orientation to immediate obstacle avoidance. The approach where the current angular velocity is oriented to the Left will be further examined, however, a similar approach can be used in the opposite case if the current angular velocity is oriented to the Right. The following rules delineate the robot’s response based on rangefinder observations:

- a) Rule for Left angular velocity with Close central detection. The robot’s output will be to continue turning to the Left to quickly navigate around an impending obstacle. However, an exception arises when the left observer also detects an obstacle as Close and the right observer detects Not Detected (Fig. 9 (a)). In such a case, the output switches to Right to exploit potential open space on the right side and to avoid being ensnared by a potential trap to the Left.
- b) Rule for Left angular velocity with both sides Close. If obstacles are detected as Close by both the left and right observers (Fig. 9 (b)), the output remains Left. This decision aims to enhance the robot’s angular velocity to swiftly circumvent closely positioned obstacles.
- c) Rule for Left angular velocity with variable side detections. When obstacles are detected by the left and right observers at varying proximity (Fig. 10), the output is dictated by which side is detected as closer. If the left side is closer (Fig. 10 (a)), the robot will turn to the Right. If the right side is closer (Fig. 10 (b)), the swarm continues turning to the Left. If detections are equal (Fig. 10 (c)), the robot maintains its current angular velocity (Straight) to avoid both reducing its turning rate (which could lead to collisions) and increasing it unnecessarily (to maintain maneuverability for unforeseen obstacles).

3.3.3.4. Rule for extreme angular velocities. The last rule within the fuzzy logic control system is designed to prioritize stability by mitigating the risk of abrupt maneuvers. If the system detects that the current angular velocity is at an extreme, either Sharp Left or

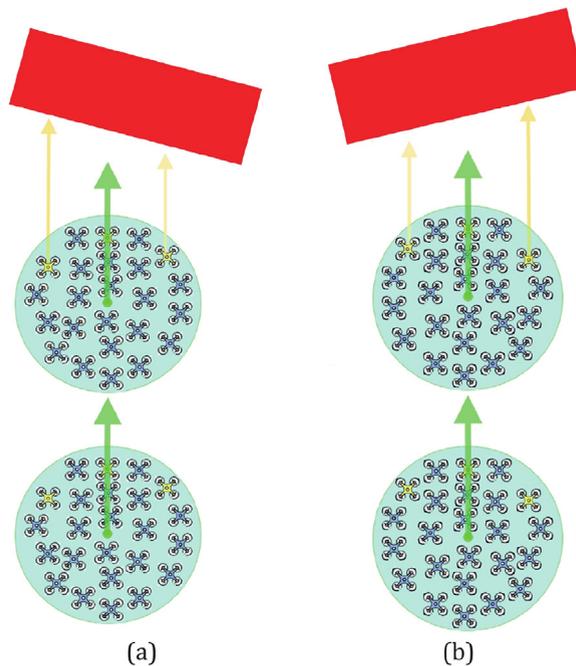


Fig. 7. Cases when Straight angular velocity with unequal side obstacle detections: (a) right obstacle is closer; (b) left obstacle is closer.

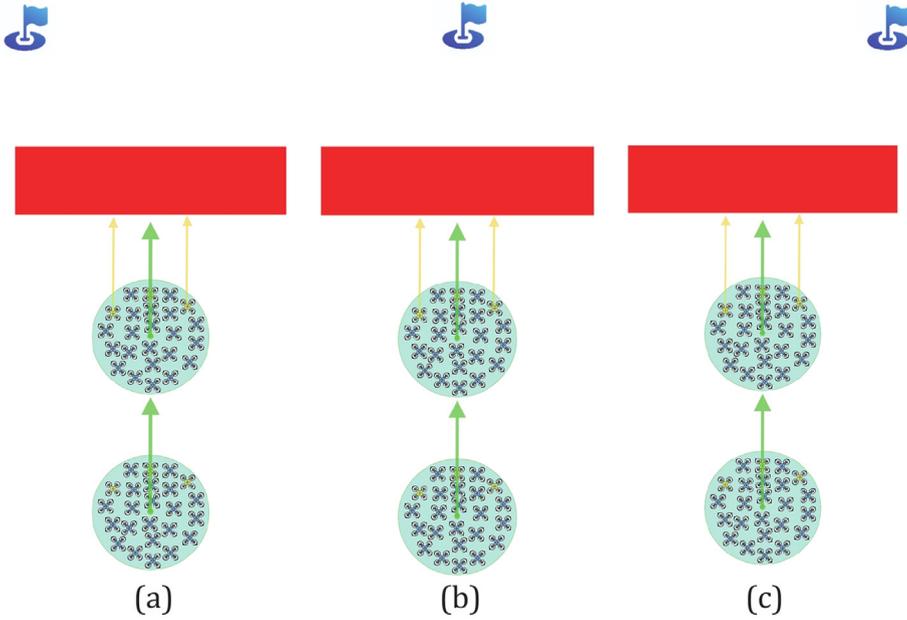


Fig. 8. Cases when Straight angular velocity with equal obstacle detections and goal side of (a) Left, (b) Straight, and (c) Right.

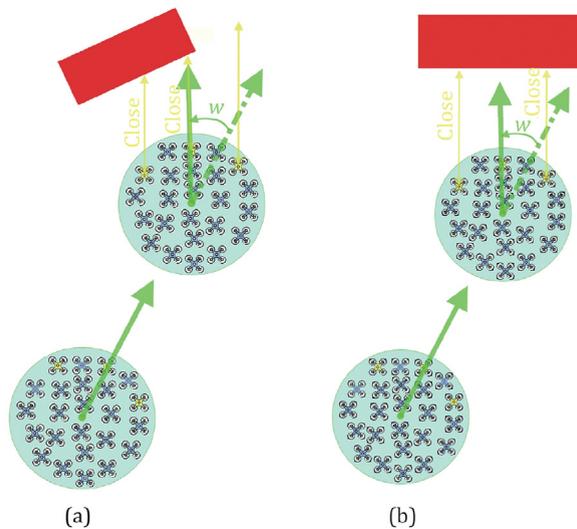


Fig. 9. Cases when (a) central and left observers detect an obstacle as Close and right observer detects Not Detected; (b) current angular velocity is Left and the obstacles are detected as Close by both the left and right observers.

Sharp Right, it automatically adjusts the output to steer the vehicle towards the opposite direction. This decision is independent of other sensor inputs such as the goal side orientation or rangefinder detections.

The peaks of the Sharp Left and Sharp Right membership functions are set at $-\pi/2 + w$ and $\pi/2 - w$, ensuring that the vehicle's angular velocity does not surpass $\pi/2$. Exceeding this threshold would command the vehicle to execute a reversal, which is both inefficient and unsafe. The use of centroid defuzzification in calculating the output moderates the rate of change in the angular velocity. As the vehicle's angular velocity nears these threshold values, the relevance of the Sharp Left or Sharp Right membership function increases, thereby reducing the acceleration of the angular velocity incrementally. This rule acts as a critical safeguard, preventing the vehicle from undertaking extreme turns that could compromise its stability and safety.

These rules enable the robot to make nuanced adjustments based on the proximity and location of obstacles, balancing the need to avoid collisions with the flexibility to adapt to new obstacles as they come within sensor_range.

In addressing scenarios where the current angular velocity is Right, the same principles apply as that with the Left angular velocity, albeit with directionality reversed. The rules that govern the response to obstacles, goal orientations, and sensor readings are mirrored to

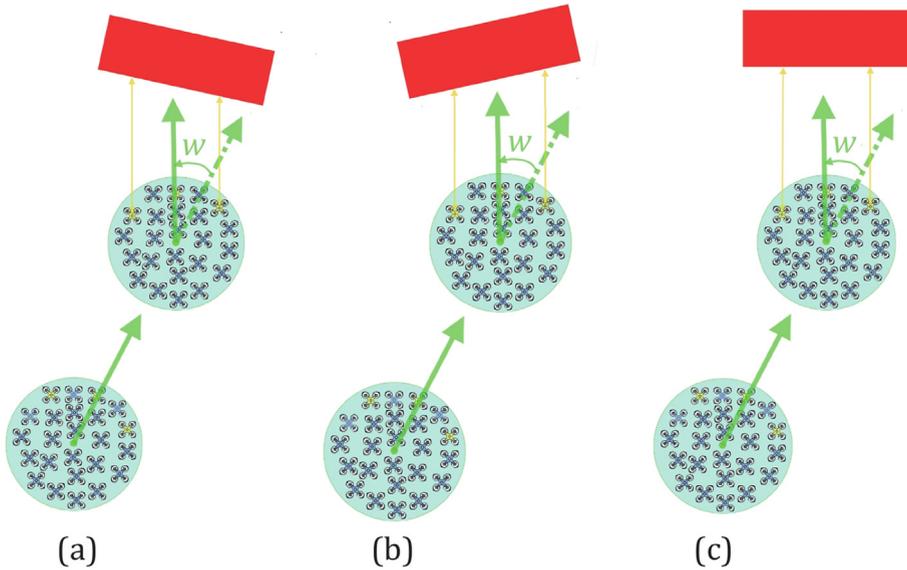


Fig. 10. Cases when the current angular velocity is Left and (a) left and right observers' detections are same to each other; (b) right observer's detection is closer than that of left observer; (c) left observer's detection is closer than that of right observer.

accommodate the rightward movement of the robot.

In conclusion, the fuzzy rules detailed above form the backbone of the fuzzy logic control system, enabling the drone to make nuanced decisions in real-time based on a variety of sensor inputs. These rules allow the system to handle complex scenarios with a level of adaptability akin to human reasoning, ensuring safe and efficient navigation. Due to the extensive number of rules involved—405 rows in total—the complete rule set is not included in this article. However, for comprehensive understanding and further analysis, the entire set of fuzzy rules has been provided as a CSV file in Appendix A.

4. Results

In this study, we present the results of implementing an intelligent obstacle avoidance algorithm for urban monitoring with autonomous drones through simulations tests, with Astana, the capital of Kazakhstan, serving as a primary case study. The algorithm's effectiveness was evaluated through various metrics including the obstacle detection accuracy, path deviation from predefined routes, and overall mission success rate in urban environments.



Fig. 11. Selected area of the city (a) as a drawn map and (b) in Google Maps.

To demonstrate the effectiveness of the algorithm, a section in the city of Astana, specifically within the Esil district, was utilized. This area is proximate to major thoroughfares such as Turkistan Street, Mangilik Yel Street, and Bukeikhanov Street. The selected sections for the algorithm simulation are depicted in Fig. 11, both on a drawn map and on Google Maps, facilitating a comparative analysis between the algorithm’s performance in simulated and real-world urban environments.

The utilized technology involved the development of a custom simulator using the Python programming language, which conducts simulations of urban scenarios with artificially created obstacles. These obstacles are represented by buildings marked in dark color on the map in Fig. 11. A total of 25 coordinated drones were employed, with their maximum angular velocity adjustments (w) regulated at 1° , 2° , 3° , 5° , 10° , 15° , 20° , and 30° , and linear velocities (V) ranging from 0.1 m/s to 10.0 m/s. To assess effectiveness, the overall success of navigating around all obstacles and the minimum number of required steps to achieve the goal were evaluated. Fig. 12 shows the number of drones that “survive”, i.e., completely avoid obstacles at different linear velocities and angular velocity adjustments.

The experimental results revealed that the number of drones that successfully avoided all obstacles and reached their destinations varied significantly with the changes in the maximum angular velocity adjustment (w) and linear velocity (V). The survival count, defined as the number of drones that successfully navigated the urban environment without collisions, showed clear dependency on both w and V . For example, when $w = 5^\circ$ and $V = 3.5$ m/s, the survival count dropped sharply, resulting in a small number of surviving drones. In contrast, with $w = 20^\circ$ and $V = 1$ m/s, the survival count was relatively stable, with a larger number of drones reaching their destinations.

Furthermore, the experiments demonstrated that higher values of w allowed for more agile maneuvers, enhancing obstacle avoidance capabilities. However, excessively high w values also led to instability and increased the risk of overshooting the target. For instance, at $w = 15^\circ$ and $V = 10$ m/s, drones exhibited a balanced ability to avoid obstacles and maintain a steady course, resulting in a higher survival count. Conversely, lower values of w (e.g., $w = 3^\circ$) resulted in inferior obstacle avoidance, leading to more collisions, especially at higher velocities.

Fig. 13 shows the number of steps required for drones to reach the target. When $w = 15^\circ$ (Fig. 13 (a)), the number of required steps decreased sharply and stabilized at a linear velocity of 1.5 m/s, indicating efficient navigation. Similarly, for $w = 30^\circ$ (Fig. 13 (b)), the number of required steps also showed a rapid decrease and stabilization when the linear velocity reached 2.0 m/s, but with slightly higher efficiency compared with that of $w = 15^\circ$. It is important to note that the delays in the figures indicate configurations where none of the drones reached the destination.

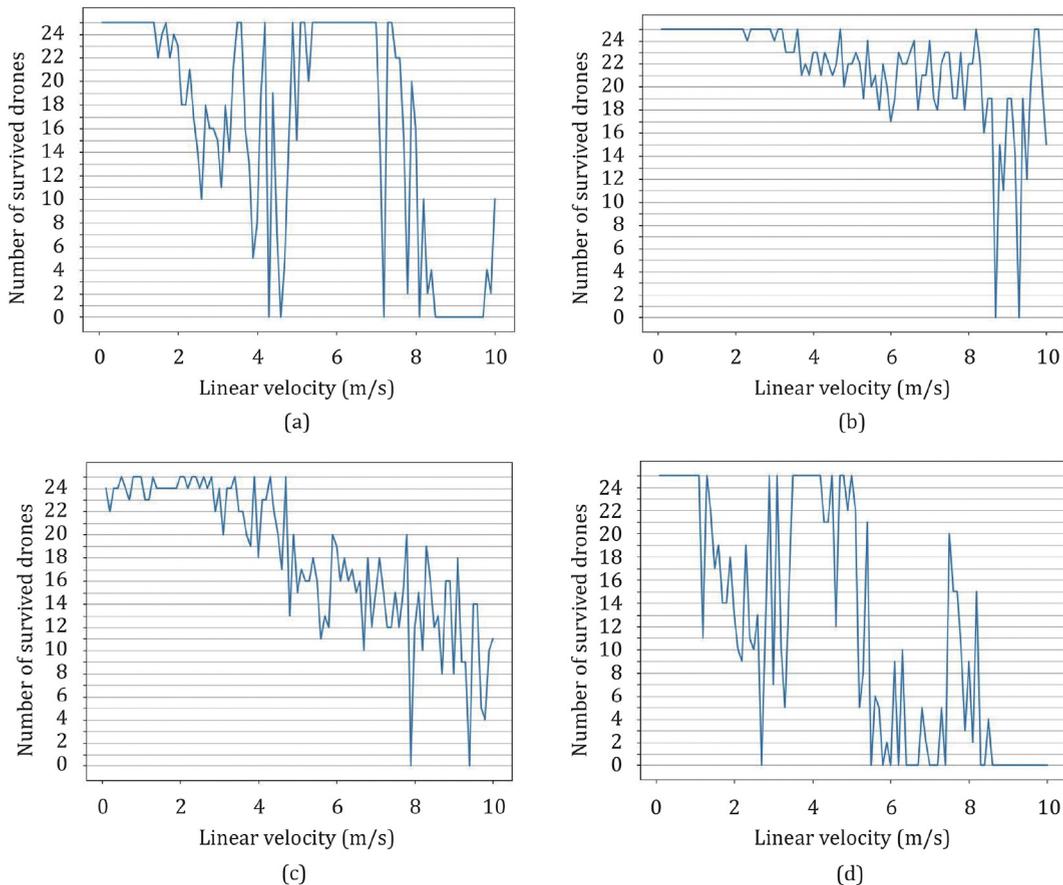


Fig. 12. Results of the numbers of the “survived” drones in the cases when (a) $w = 5^\circ$, (b) $w = 20^\circ$, (c) $w = 30^\circ$, and (d) $w = 3^\circ$.

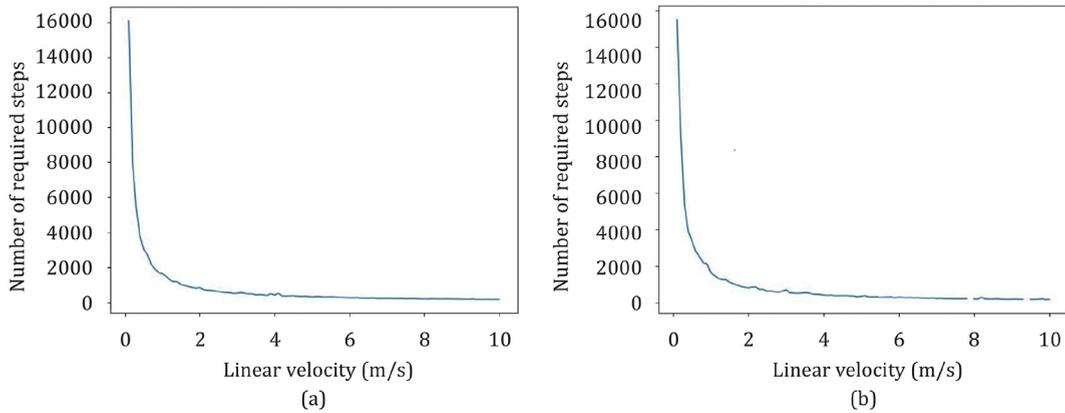


Fig. 13. Results of the number of required steps to reach the destination in the cases when (a) $w = 15^\circ$ and (b) $w = 30^\circ$.

These results indicate that as the maximum angular velocity adjustment parameter (w) increases, the number of steps required for drones to reach their target decreases sharply and then stabilizes. This suggests that higher values of w enable more efficient navigation by allowing greater agility in obstacle avoidance. However, excessively high values of w can introduce instability, as seen in Fig. 13 (b) with $w = 30^\circ$. In this configuration, while the initial steps to reach the target decrease, the higher agility leads to increased collisions. This sometimes results in scenarios where no drones reach the target, indicated by the delays in the plot. These findings underscore the importance of selecting an optimal w value that balances maneuverability and stability, ensuring effective and reliable drone coordination and obstacle avoidance.

Compared with the existing methods, such as the reactive obstacle avoidance algorithm [26], the dual-game based algorithm [27], and the hybrid swarm intelligent algorithm [28], our proposed algorithm solely relies on lidar data and inter-drone communications, contributing to lower resource consumption and energy usage by eliminating the need for camera images. Additionally, the proposed algorithm optimizes the resource usage by enabling only three drones from the swarm to be active at one moment. The system does not require central control; the drones autonomously make decisions based on the target position, avoiding the need for complex central algorithms. Moreover, the decentralized nature of the system ensures that it remains functional even if some drones crash or lose connection, continuing to operate as long as at least one drone is alive. The use of fuzzy logic further enhances the adaptability and decision-making capabilities of the drones, ensuring robust performance in dynamic urban environments.

While this study considers a flight model in 2D space to simplify initial development and testing, it is important to note that the algorithm performs well when drones are flying at the same level and maintaining 2D formation patterns, which is common in practical scenarios. This approach reduces resource usage and energy consumption while simplifying calculations. However, we recognize the importance of accounting for 3D space in urban conditions, particularly considering the height of buildings and obstacles such as power lines. Future improvements to the algorithm will incorporate 3D flight capabilities to enhance its applicability in more complex urban environments.

These results showcase the outcomes of applying a smart obstacle avoidance algorithm in monitoring urban areas using autonomous drones, with Astana, the capital of Kazakhstan, as the focal point. We assessed the algorithm's performance using simulation tests, and proved that this algorithm can be developed further. A demo video illustrating the algorithm is provided at <https://youtu.be/FKGU2zn34R0>. Additionally, Appendix A includes a ZIP file containing CSV files with detailed experimental research results and the trajectories of all experiments.

5. Conclusions

In conclusion, the development and implementation of the intelligent obstacle avoidance algorithm mark a significant advancement in the realm of urban monitoring utilizing autonomous technologies, particularly drone swarms. Through the comprehensive simulations conducted in this study, we have demonstrated the algorithm's efficacy in enhancing the obstacle detection precision and navigation accuracy within diverse urban landscapes. The algorithm's success lies in its ability to address the inherent challenges of coordinating drone movements amidst urban obstacles while optimizing energy efficiency.

Specifically, the incorporation of three key components—calculation of virtual leader position, identification of observers within the swarm, and utilization of fuzzy logic for angular velocity calculation—has enabled the algorithm to dynamically adapt to varying urban conditions. The fuzzy logic subsystem, in particular, plays a crucial role in facilitating real-time decision-making by adjusting the angular velocity based on sensor inputs and goal direction. This dynamic adjustment allows the drone swarm to navigate in complex environments with a degree of uncertainty, ensuring fluid motion and effective obstacle avoidance while keeping focusing on mission objectives.

Experimental results indicate that higher values of angular velocity adjustment (w) enhance the drones' obstacle avoidance capabilities, though excessively high values can cause instability. Optimal performance was observed at specific values of w and linear velocity (V), such as $w = 15^\circ$ and $V = 10$ m/s, where the balance between agility and stability was achieved, leading to a higher survival

count and efficient navigation. The study also revealed that the algorithm can successfully adapt to scenarios where drones become unavailable, maintaining the formation and mission success by dynamically recalculating observer roles.

As a concluding remark for the “fuzzy rules” subsection, it is imperative to underscore the pivotal role of these rules in providing a framework for a fuzzy logic system that enables agile navigation in dynamic and unpredictable urban settings. By incorporating fuzzy logic, the algorithm ensures the autonomous drones can adapt swiftly to changing environmental conditions, thereby enhancing their overall effectiveness in urban monitoring tasks. This flexible approach to decision-making underscores the algorithm’s potential for safe and efficient deployment in real-world urban monitoring applications.

This adaptability, enabled by real-time decision-making based on lidar detections, is critical for effective urban monitoring, allowing the drones to react promptly to unforeseen obstacles and environmental changes. It underscores the algorithm’s robustness in handling a variety of urban scenarios.

The algorithm’s reliance on lidar data and inter-drone communications, as opposed to camera images, significantly reduces resource consumption and energy usage, allowing for faster decision-making and simpler, cheaper devices. The system’s decentralized nature ensures continuous operation even if some drones crash or lose connection, and its efficiency is further enhanced by using fuzzy logic for decision-making.

While the current study focuses on a 2D flight model to simplify the initial development and testing, the algorithm performs well for drones flying at the same level and maintaining 2D formation patterns, which is common in practical scenarios. However, recognizing the importance of 3D flight in urban environments, future improvements will incorporate 3D flight capabilities to enhance the algorithm’s applicability in more complex settings.

Overall, this study highlights the potential of the intelligent obstacle avoidance algorithm in transforming urban monitoring practices with autonomous drone swarms. The demonstrated efficiency and adaptability of the algorithm set a solid foundation for future advancements, promising enhanced performance and broader applicability in increasingly complex urban environments.

Declaration of competing interest

The authors declare no conflicts of interest.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.jnlest.2024.100277>.

References

- [1] J. Vodák, D. Šulyová, M. Kubina, Advanced technologies and their use in smart city management, *Sustainability* 13 (10) (May 2021) 5746:1–20.
- [2] R. Petrolo, V. Loscri, N. Mitton, Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms, *T. Emerg. Telecommun. T.* 28 (1) (Jan. 2017) e2931:1–21.
- [3] F. Outay, H.A. Mengash, M. Adnan, Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: recent advances and challenges, *Transport. Res. A: Pol.* 141 (Nov. 2020) 116–129.
- [4] J.I. Hernández-Vega, E.R. Varela, N.H. Romero, C. Hernández-Santos, J.L.S. Cuevas, D.G.P. Gorham, Internet of things (IoT) for monitoring air pollutants with an unmanned aerial vehicle (UAV) in a smart city, in: F.T. Guerrero, J. Lozoya-Santos, E.G. Mendivil, L. Neira-Tovar, P.G.R. Flores, J. Martín-Gutierrez (Eds.), *Smart Technology*, Springer, Cham, Germany, 2018, pp. 108–120.
- [5] A. Gohari, A.B. Ahmad, R.B.A. Rahim, et al., Involvement of surveillance drones in smart cities: a systematic review, *IEEE Access* 10 (May 2022) 56611–56628.
- [6] M. Hossain, A. Hossain, F.A. Sunny, A UAV-based traffic monitoring system for smart cities, in: *Proc. of Intl. Conf. on Sustainable Technologies for Industry 4.0*, Dhaka, Bangladesh, 2019, pp. 1–6.
- [7] J.J. Roldán, P. Garcia-Aunon, E. Peña-Tapia, A. Barrientos, SwarmCity project: Can an aerial swarm monitor traffic in a smart city, in: *Proc. of IEEE Intl. Conf. on Pervasive Computing and Communications Workshops*, Kyoto, Japan, 2019, pp. 862–867.
- [8] P. Garcia-Aunon, J.J. Roldán, A. Barrientos, Monitoring traffic in future cities with aerial swarms: Developing and optimizing a behavior-based surveillance algorithm, *Cogn. Syst. Res.* 54 (May 2019) 273–286.
- [9] C.J. de Frias, A. Al-Kaff, F.M. Moreno, Á. Madridano, J.M. Armingol, Intelligent cooperative system for traffic monitoring in smart cities, in: *Proc. of IEEE Intelligent Vehicles Symposium*, Las Vegas, USA, 2020, pp. 33–38.
- [10] P. Gogoi, J. Dutta, R. Matam, M. Mukherjee, An UAV assisted multi-sensor based smart parking system, in: *Proc. of IEEE Conf. on Computer Communications Workshops*, Toronto, Canada, 2020, pp. 1225–1230.
- [11] M.S. Shirazi, A. Patooghy, R. Shisheie, M. Haque, Application of unmanned aerial vehicles in smart cities using computer vision techniques, in: *Proc. of IEEE Intl. Smart Cities Conf.*, Piscataway, USA, 2020, pp. 1–7.
- [12] A. Beg, A.R. Qureshi, T. Sheltami, A. Yasar, UAV-enabled intelligent traffic policing and emergency response handling system for the smart city, *Pers. Ubiquit. Comput.* 25 (1) (Feb. 2021) 33–50.
- [13] L.-Y. Chen, H.S. Huang, C.-J. Wu, Y.T. Tsai, Y.-S. Chang, A LoRa-based air quality monitor on unmanned aerial vehicle for smart city, in: *Proc. of Intl. Conf. on System Science and Engineering*, New Taipei, China, 2018, pp. 1–5.
- [14] G.P. Mayuga, C. Favila, C. Oppus, E. Macatulad, L.H. Lim, Airborne particulate matter monitoring using UAVs for smart cities and urban areas, in: *Proc. of IEEE Region 10 Conf.*, Jeju, Korea (South), 2018, pp. 1398–1402.
- [15] P. Yadav, T. Porwal, V. Jha, S. Indu, Emerging low-cost air quality monitoring techniques for smart cities with UAV, in: *Proc. of IEEE Intl. Conf. on Electronics, Computing and Communication Technologies*, Bangalore, India, 2020, pp. 1–6.
- [16] Z.-W. Hu, Z.-X. Bai, Y.-Z. Yang, Z.-J. Zheng, K.-G. Bian, L.-Y. Song, UAV aided aerial-ground IoT for air quality sensing in smart city: Architecture, technologies, and implementation, *IEEE Network* 33 (2) (Mar.–Apr. 2019) 14–22.
- [17] J.-H. Gao, Z.-W. Hu, K.-G. Bian, X.-Y. Mao, L.-Y. Song, AQ360: UAV-aided air quality monitoring by 360-degree aerial panoramic images in urban areas, *IEEE Internet Things* 8 (1) (Jan. 2021) 428–442.
- [18] J. Seo, L. Duque, J. Wacker, Drone-enabled bridge inspection methodology and application, *Automat. Constr.* 94 (Oct. 2018) 112–126.

- [19] A.B. Barreto, R.A.T. Santos, P.E.U. De Souza, M. Abrunhosa, A. Dominice, J.D. De Souza, Smart-grid assets inspections—enabling the smart cities infrastructure, in: Proc. of Intl. Conf. on Computational Science and Computational Intelligence, Las Vegas, USA, 2018, pp. 531–536.
- [20] Y.-F. Pan, F. Liu, J. Yang, et al., Broken power strand detection with aerial images: A machine learning based approach, in: Proc. Of IEEE Intl. Smart Cities Conf., Piscataway, USA, 2020, pp. 1–7.
- [21] W.-D. Wu, M.A. Qurishee, J. Owino, I. Fomunung, M. Onyango, B. Atolagbe, Coupling deep learning and UAV for infrastructure condition assessment automation, in: Proc. of IEEE Intl. Smart Cities Conf., Kansas City, USA, 2018, pp. 1–7.
- [22] R. Roberts, L. Inzerillo, G. Di Mino, Using UAV based 3D modelling to provide smart monitoring of road pavement conditions, *Information* 11 (12) (Dec. 2020) 568:1–24.
- [23] T.P. Latha, K.N. Sundari, S. Cherukuri, M.V.V.S.V. Prasad, Remote sensing UAV/drone technology as a tool for urban development measures in APCRDA, in: Proc. of Intl. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Enschede, The Netherlands, 2019, pp. 525–529.
- [24] J.J. Roldán-Gómez, P. Garcia-Aunon, P. Mazariegos, A. Barrientos, SwarmCity project: Monitoring traffic, pedestrians, climate, and pollution with an aerial robotic swarm, *Pers. Ubiquit. Comput.* 26 (4) (Mar. 2020) 1151–1167.
- [25] H.-R. Xu, L.-Z. Wang, W. Han, et al., A survey on UAV applications in smart city management: Challenges, advances, and opportunities, *IEEE J -STARS* 16 (Sept. 2023) 8982–9010.
- [26] F. Azevedo, J.S. Cardoso, A. Ferreira, T. Fernandes, M. Moreira, L. Campos, Efficient reactive obstacle avoidance using spirals for escape, *Drones* 5 (2) (Jun. 2021) 51:1–26.
- [27] Y. Lin, Z.-Y. Na, Z.-L. Feng, B. Lin, Y. Lin, Dual-game based UAV swarm obstacle avoidance algorithm in multi-narrow type obstacle scenarios, *EURASIP J. Adv. Sig. Pr.* 2023 (1) (Nov. 2023) 118:1–21.
- [28] C.-Y. Gao, J.-F. Ma, T. Li, Y.-L. Shen, Hybrid swarm intelligent algorithm for multi-UAV formation reconfiguration, *Complex Intell. Syst.* 9 (2) (Apr. 2023) 1929–1962.



Didar Yedilkhan is an associate professor with the Department of Computer Engineering, Astana IT University, Astana, Kazakhstan. He received the B.S. degree in information systems from Al-Farabi Kazakh National University, Almaty, Kazakhstan in 2011, the M.S. degree in information systems from International Information Technology University, Almaty, Kazakhstan in 2013, and the M.S. degree in business analytics from University College London, London, UK in 2019. He earned his Ph.D. degree in computer systems and software engineering from International Information Technology University in 2016. His main research interests include data science, machine learning, intellectual systems, business analytics, process mining, business process management, and data engineering.



Abzal E. Kyzrkanov received the B.S. degree in computer science from Al-Farabi Kazakh National University in 2014 and the M.S. degree in cyber security from ITMO University, St. Petersburg, Russia in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer and Software Engineering, L. N. Gumilyov Eurasian National University, Astana, Kazakhstan. He is also a senior lecturer with the Department of Computer Engineering, Astana IT University. His main research interests include fuzzy logic, artificial intelligence, swarm robotics, multi-agent systems, formation control, machine learning, mobile robots, and unmanned aerial vehicles (UAVs).



Zarina A. Kutpanova was born in Semey, Kazakhstan in 1989. She received the B.S. degree with honors in computer science from Semey State University, Semey, Kazakhstan in 2011 and the M.S. degree from Kazakh Humanitarian Juridical Innovative University, Semey, Kazakhstan in 2013. She is currently pursuing the Ph.D. degree with the School of Computer Science, Gumilyov Eurasian National University, Astana, Kazakhstan. Her main research interest focuses on obstacle avoidance swarm of UAVs which follow one by one.



Shadi Aljawarneh is a full professor with the Department of Software Engineering, Jordan University of Science and Technology, Irbid, Jordan. His research interests include e-learning, cloud computing, bioinformatics, and information and communications technology fields.



Sabyrzhan K. Atanov received the B.S. degree in space radio communications from Moscow Technical University of Communications and Informatics, Moscow, Russia in 1979, and the Ph.D. degree in theoretical electrical engineering from Moscow Power Engineering Institute, Moscow, Russia in 1989. In 2010, he received a Doctor of Engineering degree in the specialty of mathematical software support of computing machines, complexes, and computer networks. He is currently a professor with the Department of Computer Science and Engineering, Gumilyov Eurasian National University. He has experience in the development of robotic systems, programming in high- and low-level languages, and numerical modeling. He has a great deal of experience in the management of state budget projects and initiative research. For the past 5 years, he has been working on theoretical and practical issues of artificial intelligence. He has extensive experience with machine learning algorithms, pattern recognition, and microcontroller system design. He has published more than 30 scientific papers.