Robotica (2026), **44**, pp. 1573–1607
doi:10.1017/S0263574726103725

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Human-swarm interaction for formation/flying control by enhanced vision-based hand gesture recognition system

Fatna Bent Ennebi Benabbou, Belkacem Khaldi and Sidi Mohammed Benslimane

LabRI-SBA Lab., Ecole Superieure en Informatique, Sidi Bel Abbes, Algeria
Corresponding author: Fatna Bent Ennebi Benabbou; Email: f.benabbou@esi-sba.dz

Received: 23 August 2025 UTC; **Revised:** 28 May 2026 UTC; **Accepted:** 7 June 2026 UTC;
First published online: 8 July 2026 UTC

Keywords: human-swarm interaction; gesture recognition; drone formations; Bidirectional Gated Recurrent Unit (BiGRU); intuitive interface

Abstract

This study introduces an improved gesture-based human–swarm interaction control system for drone swarms that unifies motion and formation control while addressing the critical challenge of neglecting the impact of faulty robots on task performance. Unlike conventional approaches that treat these functionalities separately, our integrated framework enables robust task execution in challenging environments. Using 26 hand gestures (18 formations and 8 flight commands) processed by a variety of deep learning models, including convolutional neural networks (e.g., ResNet101 and MobileNetV2) and recurrent neural network-based approaches such as long short-term memory (LSTM), bidirectional LSTM, gated recurrent unit (GRU), and bidirectional GRU (BiGRU), the system provides comprehensive yet accessible control, notably through the BiGRU network with 99.4% accuracy in real-time gesture recognition tasks. Deployment testing conducted using the CrazyFlyt simulation platform demonstrated the system's robustness in maintaining user-intended formations with low positional error and stable convergence times, even in the presence of sensor noise and faulty robots. Statistical analysis revealed highly significant main and interaction effects ($p < 0.05$) for swarm scale and scenario on performance metrics, indicating that the impact of swarm size is dynamically contingent upon environmental conditions. Post hoc evaluations predominantly highlighted significant performance degradation in faulty-robot scenarios compared to normal or noisy conditions, with swarm scale critically modulating these effects.

1. Introduction

Swarm robotics represents a groundbreaking advancement in robotics, where numerous autonomous agents work together to achieve complex tasks without central control [1, 2]. This concept draws inspiration from natural phenomena such as ant colonies and bee swarms, demonstrating remarkable efficiency in tasks such as foraging and territory defence. Despite the potential of these robotic systems to perform complex tasks, they often face challenges in effectively integrating human input into collaborative operations. This gap underscores the critical importance of human–swarm interaction (HSI) as a specialised domain within human–robot interaction (HRI), aimed at overcoming these limitations and fostering more efficient collaboration between humans and swarms [3, 4]. Although HSI offers flexibility in coordinating with multiple robots through interpreters compared to HRI, it presents significant challenges to achieving intuitive control while ensuring robustness, scalability, and safety. Traditional solutions for HSI leverage intermediary devices such as joysticks [5] or virtual reality (VR) [6] headsets to enable precise command execution and immersive experiences for certain users. For instance, a joystick enables straightforward navigation and direction of drone swarm movements, while VR headsets allow users to interact with the robots through gestures and spatial navigation [7], offering an enhanced sense of control. However, these devices often pose accessibility limitations. Individuals with physical disabilities,

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

such as those lacking dexterity or mobility, or those with special needs, such as deafness, dumbness, or blindness, may find these intermediary technologies difficult to use effectively [8]. These limitations highlight the urgent need for more inclusive solutions that cater to a broader range of users.

One HSI strategy is gesture-based interactions, which presents a promising approach for enhancing interaction with robotic swarms [9, 10]. This method leverages advancements in visual-based hand gesture recognition facilitated by convolutional neural networks (CNNs) and recurrent neural networks (RNNs), significantly improving both accuracy and processing speed [11–14]. This approach aligns with broader trends in HRI that utilise uncertainty-aware AI, such as the implementation of advanced deep learning within humanoid and assistive robotics to produce adaptable actions based on recognised human intentions [15–17]. However, despite these enhancements, current systems are often limited to controlling individual robots, failing to effectively manage the collective behaviour of swarms [18, 19]. While this approach has shown promise in enhancing human–robot collaboration, existing implementations often suffer from three critical limitations: (1) the independent treatment of motion and formation control, which reduces coordination efficiency and adaptability in dynamic environments. (2) the assumption of ideal, homogeneous robot behaviour, neglecting faults or degraded agents arising from actuator failures, sensor errors, or communication issues, and (3) the limited integration of high-accuracy deep learning models, which remain focused on single-robot tasks rather than collective swarm control.

Recent advances in adaptive and robust control strategies may offer potential solutions to these critical challenges. Indirect self-tuning [20] and constrained adaptive sliding-mode controllers [21, 22], for example, can automatically adjust to uncertainties, directly addressing the issue of neglected agent faults. Similarly, multi-sensor fusion using extended Kalman filters improves state estimation robustness against noise [23], while optimised trajectory planning facilitates better coordination in dynamic environments [24]. However, despite their potential to enable fault-tolerant, adaptive systems, the application of these robust strategies remains largely confined to non-swarm robotics studies.

To address these limitations, particularly the decoupling of motion and formation control, as well as the oversight regarding the impact of faulty agents, we propose a unified hand gesture framework. This system facilitates seamless task execution and ensures robust adaptability in challenging environments, such as those characterised by sensor noise or drone malfunctions. The system utilises eighteen distinct hand gestures for swarm formations and an additional eight for navigation commands. This configuration balances complexity with simplicity, enabling intuitive operator interaction. To ensure robust performance, we evaluate multiple deep learning architectures for real-time gesture recognition, comparing CNN-based models such as ResNet101 and MobileNetV2 with RNN variants such as LSTM, BiLSTM, GRU, and BiGRU. The optimised BiGRU model achieves a very acceptable 99.40% accuracy in gesture recognition, demonstrating its effectiveness across diverse operational scenarios, including normal conditions, environments with sensor noise, and situations involving drone failures.

Deployed testing conducted using the CrazyFlyt[1] simulator demonstrates the system's scalability across swarm sizes of 7, 15, and 25 drones engaged in different formations/navigation tasks. Notably, most of the desired tasks consistently exhibit superior efficiency with minimal variability in task completion time and positional accuracy, particularly when scaling to larger swarms of 25 drones. The system also exhibits resilience under adversities induced by sensor noise, maintaining formation integrity despite sensor degradation. Furthermore, it preserves task performance even when up to 20% of drones experienced simulated failures, highlighting its ability to adapt to agent failures without compromising overall swarm functionality.

The remainder of this paper is organised as follows: Section 2 reviews the relevant literature and establishes the theoretical context for our study. Section 3 details the vision-based hand gesture recognition framework, including the MediaPipe-extracted dataset, the implementation of diverse CNN and RNN architectures, and the corresponding training results. Section 4 discusses the development of the

[1] https://github.com/jjshoots/CrazyFlyt/tree/master

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

swarm controller and the integrated formation and flight control logic designed to manage system uncertainties. Section 5 provides an in-depth analysis of the experimental findings, specifically evaluating system robustness against sensor noise and drone failures. Finally, Section 6 concludes the paper with a summary of key findings and potential directions for future research.

2. Related work

From a control-theoretic perspective, swarm robotics formation is often achieved through decentralised control strategies, where each agent relies only on local interactions, collectively generating global shapes without centralised supervision. Hsieh et al. [25] presented a decentralised control framework for swarm formation where robots use local interaction laws to form and maintain geometric shapes, offering scalability and robustness but lacking human-in-the-loop interaction or user command interpretation. Sanchez et al. [26] developed a gesture-based human–robot interaction system for safe quadrotor navigation using a bracelet with inertial measurement units (IMU) and electromyographic (EMG) sensors to translate user gestures into semi-autonomous flight commands with safety features. Tests on a real quadrotor demonstrate effective recognition and control. However, the method is limited to single-drone use and needs specialised wearable hardware, unlike vision-based or swarm-scalable approaches.

Deep learning techniques, especially CNNs and RNNs, have made a big difference in the fields of gesture recognition and HSI. This section looks at recent studies, focusing on the methods used, the datasets used, and the accurate results.

Gesture recognition systems have made a lot of progress lately, especially in robotics and wearable technologies. Huang et al. [27] used a data glove-based gesture recognition system to map hand motions to hexacopter commands in the CoppeliaSim simulation platform, achieving 84.3% overall accuracy (94% for static gestures, 76.1% for dynamic ones), but requiring wearable hardware and showing poorer performance on dynamic gestures. While Kakish et al. [12] came up with a way to control robotic swarms using a series of hand gestures. They enabled decentralized control by combining a lightweight CNN with robot operating system (ROS2). This was proven through simulations and physical experiments, and it worked well for both gesture recognition and changing swarm behaviour. Li et al. [28] present a natural interaction control method for unmanned swarm systems utilising gesture recognition. It enhances the YOLOv5 algorithm by incorporating a DIoU loss function to optimise real-time performance and creates a gesture database for control. Testing on the AirSim platform shows a 97% accuracy rate for recognising gestures and a response time of 0.027 s. This shows that you can effectively control unmanned swarms using gesture-based interaction. Yeh et al.'s paper [19] talks about a gesture recognition system for controlling an unmanned aerial vehicle (UAV) that uses deep learning. The system is very accurate, with 99.54% on the training set and 99.17% on the testing set. This is thanks to LSTM neural networks. It makes control more responsive by using multiple threads to process data, which lowers latency and raises frame rates. This makes it a more intuitive and user-friendly way to control a UAV than traditional methods. Wang et al. [14] describe a system for visual communication between a drone and its operator that uses static gesture recognition. It uses an elliptical skin colour detection model and Residual Network ResNet34 to get features, and it gets stable control by using calibrated commands. The system was tested with a Tello drone and showed an average accuracy of 90.43% in recognising gestures, which shows that it could be useful in real life. Khaksar et al. [29] write about a system that lets you control a drone with hand gestures using the MediaPipe Hands algorithm. It achieved a classification accuracy of 96.25% for eight static gestures, showing that it could recognise gestures well. The system makes things easier to get to without needing special equipment, but it does deal with problems like Z-axis instability. In general, it shows how hand-gesture recognition could make flying a drone easier. Khen et al. [30] intend to develop an easy-to-use control interface with wearable devices and machine learning (ML), which will allow users with different skill levels to manage swarms of drones. They test various ML models such as K-nearest neighbours (KNN), support vector machines (SVM), and random forest on motion capture data, attaining the highest accuracy of 84% with Naive Bayes in the former dataset and 74% with the Ensemble model in the latter.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Huu et al. [11] describe a MediaPipe and LSTM-based framework to detect and track hand gestures. The average accuracy reached by this system is 93.3% and it is conceived for VR, Medical Monitoring, and Smart Devices control. It successfully resolves nine different gestures. Zafer et al. [31] suggest a novel method for controlling a quadruped robot using hand gestures, employing a stacked convolutional BiLSTM neural network. This approach, integrated with ROS and Gazebo, aims to enhance human–robot interaction (HRI) in critical scenarios such as search and rescue operations. The system uses computer vision to interpret gestures in real-time, mapping them to specific robot commands, and demonstrates potential through simulation-based experiments. Kawatsu et al. [18] explore the use of CNN for recognising gestures to control robots. The authors use transfer learning with the ImageNet dataset to train their model on a smaller custom dataset of about 50,000 images, achieving over 99% accuracy in gesture recognition. They also investigate using an LSTM layer for sequence-based gesture recognition, achieving around 80% accuracy with a smaller dataset. The system is applied to control an unmanned ground vehicle, demonstrating its practical application in robotic control. The paper by Kučera et al. [13] evaluates LSTM and GRU neural networks for gesture recognition using motion-capture data, achieving high accuracy. It highlights the efficiency of GRU networks, which are simpler and faster to train than LSTMs, making them suitable for real-time applications in HRI.

While the above-mentioned works demonstrate, in general, high recognition performance for single UAVs or basic swarm control, they often neglect formation shaping and robustness under non-ideal conditions. Furthermore, despite the proven suitability of RNNs for gesture recognition, both their integration into unified control frameworks and the analysis of noise or faulty agent impacts on collective performance remain largely unexplored. To quantify these research gaps, Table I compares the proposed approach with existing literature. Our method outperforms prior studies in terms of architectural diversity, model performance, and control versatility, specifically by explicitly incorporating the effects of non-ideal agent behaviour.

Particularly, in contrast to Yeh et al. [19], who proposed a single LSTM-based system achieving 99.54% training and 99.17% testing accuracy, our approach involves a comprehensive evaluation of multiple architectures – LSTM, BiLSTM, GRU, and BiGRU for RNN, as well as ResNet101 and MobileNetV2 for CNN. Each model was trained and validated independently on two diverse custom datasets. Among them, the BiGRU model demonstrated the best performance, achieving an overall accuracy of 99.40%. While Yeh et al. focused on frame rate optimisation using multithreading, our methodology emphasises both high recognition performance and inclusivity. Specifically, our system is designed for intuitive gesture-based swarm formation and motion control instead of single-drone motion control, with practical relevance for people with limited mobility or special needs. This broader applicability and robustness in human–swarm interaction scenarios make our solution more adaptable and impactful.

Following the review of related works, and as summarised in Table I, our approach distinguishes itself through both technical rigour and practical innovation. We trained and evaluated multiple deep learning architectures – LSTM, BiLSTM, GRU, and BiGRU (RNN), as well as ResNet101 and MobileNetV2 (CNNs) – on two carefully designed custom datasets. Among them, BiGRU achieved the best performance with an accuracy of 99.40%, which is highly competitive compared to existing works such as Yeh et al. (99.17%) and Kawatsu et al. (over 99%). Similarly, Yeh et al. reached very high accuracy (99.54% training, 99.17% testing) using LSTM for single UAV control with an emphasis on latency optimisation. In comparison, our approach achieves comparable accuracy while extending functionality to real-time gesture-based swarm formation and flight control, thereby supporting more complex coordination tasks and inclusive HSI.

Unlike earlier studies that mostly focus on gesture recognition for controlling just a single robot or focus solely on moving drone swarms around, our approach breaks new ground by allowing users to intuitively control both how a swarm flies and how it shapes formations – all in real time, simply by using natural hand gestures. This means that individuals can not only direct drone movements but also arrange themselves in space, making interactions smoother, more versatile, and accessible. It supports natural and intuitive HSI, making it well-suited for users with limited mobility or special needs.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table I. *Comparative summary of gesture-based control studies for drones and robots with formation and flight control details.*

Author(s)	Model/Method	Dataset/Platform	Accuracy	Swarm/Single	Robot type	Application domain	Control type
Kawatsu et al. [18]	CNN (AlexNet & ResNet-50) + LSTM	Custom dataset (50,000 images); 7 gestures	> 99% static, ∼80% sequential	Single	UGV iRobot PackBot	Military robotic control	Motion
Sanchez et al. [26]	Wearable bracelet (IMU, EMG)	Sensors data	–	Single	Quadrotor UAV	UAV navigation	Motion
Kakish et al. [12]	Lightweight CNN + ROS2	Simulation (Turtlebot3, 6 gestures)	97%	Swarm	Wheeled Turtlebots	Decentralized human-swarm control	Motion
Li et al. [28]	YOLOv5 + DIoU optimization	Custom database; AirSim	97%	Swarm	UAV	UAV swarm control	Motion
Yeh et al. [19]	LSTM (multithreading)	Custom dataset; 8 commands	99.54% (train), 99.17% (test)	Single	UAV	UAV/drone control	Motion
Huang et al. [27]	Glove-based gesture recognition	Sensors data; CoppeliaSim	84.3% overall (94% static, 76.1% dynamic)	Single	Hexacopter UAV	UAV virtual interaction and manipulation	Motion
Wang et al. [14]	Skin detection + ResNet34	Tello platform	90.43%	Single	Tello drone	Visual communication	Motion

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table I. *Continued*

Author(s)	Model/Method	Dataset/Platform	Accuracy	Swarm/Single	Robot type	Application domain	Control type
Khaksar et al. [29]	MediaPipe + SVM/ANN	8 gesture dataset	96.25%	Single	DJI Tello	Drone control using hand gestures	Motion
Khen et al. [30]	ML (KNN, SVM, RF, NB, Ensemble)	Motion capture (5 gestures)	84% (Naive Bayes)	Swarm	UAV	Wearable-based swarm control	Motion
Huu et al. [11]	MediaPipe + LSTM	Custom dataset (9 gestures)	93.3%	Single	VR/medical devices	Smart device and VR control	Motion
Zafer et al. [31]	Stacked Conv-BiLSTM	ROS + Gazebo simulation	–	Single	Quadruped (Spot Robot)	Human–robot interaction, search and rescue	Locomotion
Kučera et al. [13]	LSTM vs GRU	Motion capture (Rokoko Smartsuit Pro II)	High	Single	–	Human–robot interaction	Motion
Proposed approach	LSTM, BiLSTM, GRU, BiGRU, ResNet101, MobileNetV2	Two custom datasets; 26 gestures; CrazyFlyt simulator	**99.40% (BiGRU)**	Swarm	Quadrocopter	Gesture-based swarm flight and formation control (inclusive for special needs)	Motion + Formation

Table II. *Implementation specifications for gesture RNN and CNN-based recognition architectures.*

Category	Architecture	Impl. summary	Parameter	Value
RNN	LSTM, BiLSTM, GRU, BiGRU	Three stacked recurrent layers utilizing MediaPipe hand keypoints; includes Flatten and 26-unit Softmax layers	Layers (RNN/Dense)	3/2
			Hidden units	128/256/128
			Dropout rate	0.3
			Activation (RNN/Dense)	tanh/ReLU
			Seq. length (frames)	20
			Training batch size	64
CNN	ResNet101, MobileNetV2	Frame-based spatial extraction via pre-trained backbones; utilizes GAP and unit Dense layers	Model backbone	ResNet/ MobileNet
			GAP layer	Enabled
			Dense layer units	1024
			Dropout rate	0.5

This formation shaping and directional navigation addresses more complex coordination tasks than those covered in prior work. Additionally, while some works such as Li et al. and Kakish et al. emphasised latency and physical experiments, they do not support formation control nor address accessibility and inclusivity. In contrast, our system is scalable, decentralised, and robust against environmental variations, offering a practical, accurate, and impactful solution for real-world swarm robotics applications. This combination of high recognition accuracy, expanded control capabilities, and inclusiveness sets our approach apart within the current gesture-based robotics landscape.

3. Vision-based gesture recognition models

3.1. Architecture

In this study, we implement and compare four RNN-based and two CNN-based architectures to recognise 26 hand gestures, utilising MediaPipe [32] for its ability to provide robust and reproducible hand keypoints. To facilitate a clear comparison of their structural differences, the specification details of all models are summarised in Table II. Our selection of models is motivated by the need to balance accuracy, computational cost, and real-time performance for swarm interaction. The RNN architectures, including LSTM, BiLSTM, GRU, and BiGRU, were selected for their specialised ability to learn temporal dependencies and long-term context. For the CNN architecture, we chose ResNet101 for its effective depth via residual blocks and MobileNetV2 for its computational efficiency. Model hyperparameters were determined through literature review and empirical validation to optimise the trade-off between recognition accuracy, computational cost, and real-time performance. Specifically, the three-layer recurrent stack and tanh activation were selected to ensure learning stability. To mitigate external non-parametric uncertainties – including lighting variations and visual input noise arising from user variability – Dropout layers (0.3 for RNN, 0.5 for CNN) were integrated to enhance generalisation and prevent overfitting. Furthermore, global average pooling (GAP) was implemented within the CNN backbones to reduce dimensionality and computational complexity.

3.2. Dataset description

The study involves collecting 26 distinct hand gestures for swarm formations and navigation control (representative samples are shown in Figure 1(a)). These gestures are derived from two primary categories: 18 formation gestures inspired by American Sign Language (ASL) letters (A, B, C, H, I, L, O, R, U, etc.) and numbers (1–9), chosen for their simplicity and familiarity to enable intuitive human-swarm

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

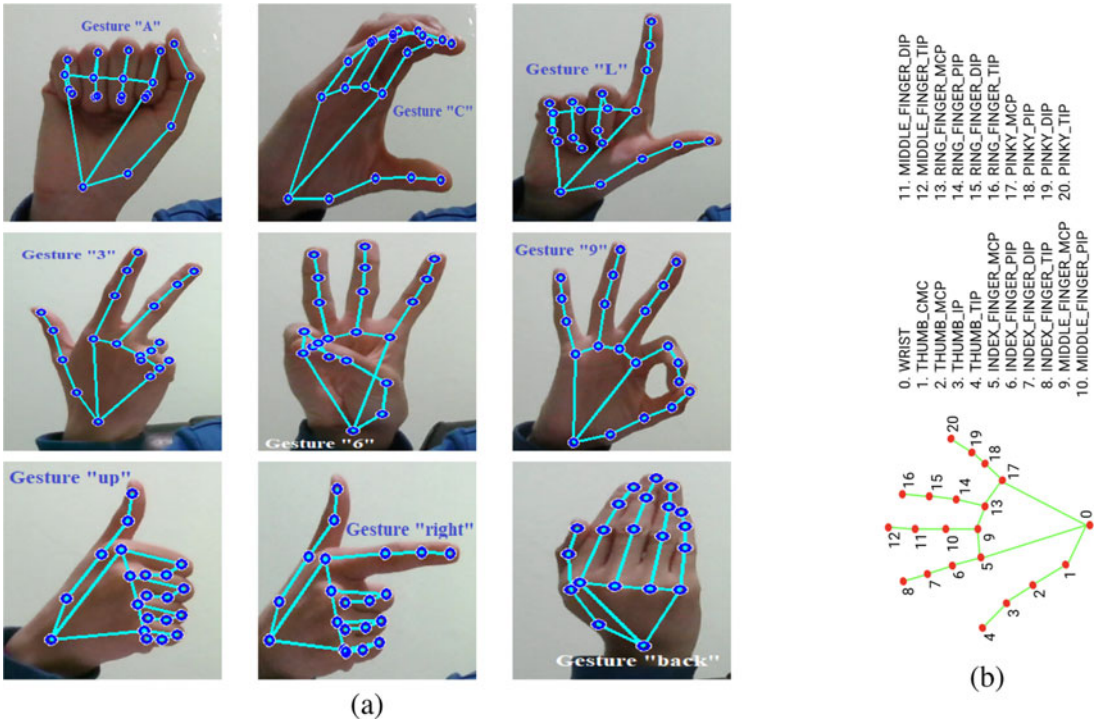


Figure 1. *Samples of hand gestures (a) with MediaPipe keypoints detection [32] (b).*

interaction. Additionally, 8 flight control gestures (up, down, left, right, backward, takeoff, stop) were selected to provide comprehensive drone motion commands, ensuring coverage of fundamental navigation and safety operations. Each gesture of the 26 is represented by 64 sequences of hand keypoint data, where each sequence comprises 20 frames for temporal information extraction. At each timestep, a set of hand keypoints is extracted using MediaPipe [32] (Figure 1(b)). For model training, approximately 33,280 frames are collected for RNNs, while 16,900 images are used for CNNs.

3.3. Training results and confusion matrices

Training Parameters: To improve the training efficiency and generalisation of the model, two key optimisation techniques were employed: a learning rate scheduler and early stopping. The learning rate scheduler begins with a rate of 0.001 and reduces this rate by a factor of 10 after every 10 epochs, allowing for larger updates initially and finer adjustments later. Additionally, an early stopping callback monitors validation loss and halts training if there is no improvement for 20 consecutive epochs, to prevent overfitting and unnecessary computations, while maintaining optimal performance on the validation set. All models were trained using the Adamax optimiser with categorical cross-entropy loss and evaluated using categorical accuracy, precision, and recall. The maximum number of training epochs was set to 200 with a batch size of 64, although training typically converged earlier due to early stopping. During training, model performance was continuously monitored on the validation set, and the best model was selected based on the minimum validation loss, in conjunction with the early stopping strategy. To ensure robust evaluation and minimise overfitting, the dataset was divided into 60% training, 20% validation, and 20% testing. This three-way split facilitates continuous performance monitoring during training and provides an unbiased evaluation on the test set.

The preprocessing pipeline is built upon MediaPipe-based hand landmark extraction, followed by normalisation of 3D coordinates and temporal sequence construction to capture dynamic gesture information. Once a gesture sequence of 20 frames is acquired, it is processed by the trained model to perform

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table III. *Results of the models.*

Architecture	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
RNN recurrent neural networks				
LSTM	97.00	96.41	96.98	96.69
BiLSTM	98.20	98.20	99.39	98.79
GRU	98.80	98.20	99.39	98.79
BiGRU	99.40	99.40	99.40	99.40
CNN conventional neural networks				
ResNet101	93.95	93.58	95.01	94.29
MobileNetV2	85.03	82.31	87.89	85.01

classification. The inference stage is computationally lightweight and enables efficient execution suitable for real-time or near-real-time applications. To accommodate users with special needs and ensure comfortable interaction, the system introduces a controlled interaction interval of 25 s between successive gesture acquisitions. This design choice allows sufficient time for users to perform gestures without time pressure, improving usability in assistive HSI scenarios. All experiments were conducted on a system equipped with an NVIDIA GPU (e.g., RTX 3060) and an Intel i7 CPU.

Performance metrics: The models are evaluated using accuracy, precision, recall, and F1-score. These metrics provide insights into the models' ability to correctly classify the 26 hand gestures. The following table summarises the results:

Table III shows that BiGRU performs the best among RNN architectures, with the highest accuracy of 99.40% and perfect precision, recall, and F1 Score, with enhanced generalisation. GRU and LSTM processes have high training accuracies but fail to do well in generalisation, as demonstrated by lower validation scores. There are signs of overfitting, as indicated by the lower validation accuracy compared to the training accuracy. The BiLSTM model achieves a strong accuracy of 98.20% and exhibits much better generalisation than both GRU and LSTM models, ranking as the best-performing model. For CNN, ResNet101 shows strong performance with an accuracy of 93.95%, while MobileNetV2 lags at 85.03%, revealing notable gaps between its training and validation accuracies. Consequently, RNNs generally excel in accuracy and performance metrics, while CNNs are effective but need some tuning to improve their performance, particularly in gesture recognition tasks.

The analysis of the confusion matrices represented in Figure 2 reveals important insights into their performance. The BiGRU model demonstrates strong performance, with a majority of predictions accurately classified and one misclassification. The model rarely confuses "R" with "U", indicating its effectiveness in distinguishing between gestures. The BiLSTM and GRU models achieve high accuracy in gesture classification, with only two misclassifications. Interestingly, R and U are misclassified, and it seems that elementary features in the data are well modeled. In contrast, the LSTM model shows a slight increase in misclassifying five instances compared to the others. An example of this is the failure to recognise "5" with "stop" in some cases, indicating possible room for improvement. These misidentifications suggest how the model can be improved. For the CNN, MobileNetV2 shows another behaviour, where more misclassifications were made. For instance, it is usually mistaken between "stop" and "1", "9" and "6" with "H", resulting in a higher error rate than RNN models. ResNet101 misclassifies less than MobileNetV2. However, it still faces challenges. In summary, the RNN models – BiGRU, BiLSTM, GRU, and LSTM – generally exhibit high accuracy with a low misclassification rate, indicating their effectiveness in distinguishing between gestures. The CNN models show more variability in performance, experiencing the most significant number of misclassifications. This analysis suggests that while RNNs are better suited for this particular task, further tuning and adjustments may enhance the performance of CNN models, especially in applications involving gesture recognition.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

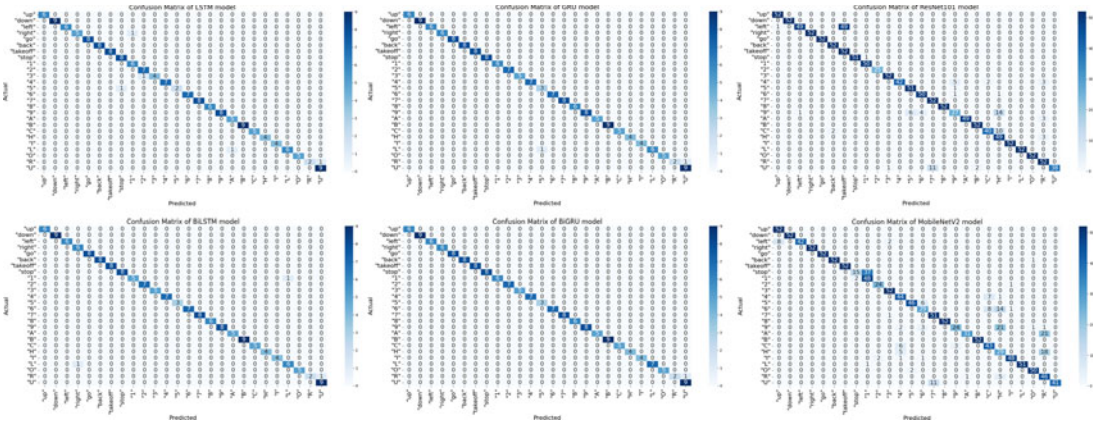


Figure 2. *(a) Confusion matrix of LSTM model (b) Confusion matrix of BiLSTM model (c) Confusion matrix of GRU model (d) Confusion matrix of BiGRU model (e) Confusion matrix of ResNet101 model (f) Confusion matrix of MobileNetV2 model.*

4. Swarm formation and flying controller

To achieve the intended formation and flight objectives for a drone swarm system, we developed a custom swarm formation and flying controller that integrates repulsive and attractive forces. This approach enables decentralised swarm control by maintaining safe inter-agent distances and guiding drones towards specific target positions. Repulsive forces are employed to prevent collisions through short-range avoidance mechanisms, while attractive forces facilitate the formation's convergence toward specified configurations. This methodology enhances scalability and robustness in dynamic environments, as demonstrated by prior research conducted by Reif and Wang [33], Gazi and Passino [34], and Kolling et al. [4].

The repulsive force of two drones i and j is calculated as follows:

$$F_{ij}^{\text{rep}} = k_{\text{rep}} \cdot \left(\frac{1}{\|\mathbf{p}_i - \mathbf{p}_j\|} - \frac{1}{r_{\text{rep}}} \right) \cdot \frac{1}{\|\mathbf{p}_i - \mathbf{p}_j\|^2} \tag{1}$$

where k_{rep} is a tunable gain controlling the strength of repulsion. The resulting repulsive force vector is oriented along the normalised direction between the two drones and is symmetrically applied in opposite directions to both agents to conserve momentum:

$$\mathbf{f}_{ij}^{\text{rep}} = F_{ij}^{\text{rep}} \cdot \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|} \quad \Rightarrow \quad \mathbf{F}_i^{\text{rep}} \mathrel{+}= \mathbf{f}_{ij}^{\text{rep}}, \quad \mathbf{F}_j^{\text{rep}} \mathrel{-}= \mathbf{f}_{ij}^{\text{rep}} \tag{2}$$

To avoid abrupt changes in velocity, a damping factor $\eta_{\text{rep}} \in [0, 1]$ is applied to the total repulsive force:

$$\mathbf{F}_i^{\text{rep}} \leftarrow \eta_{\text{rep}} \cdot \mathbf{F}_i^{\text{rep}} \tag{3}$$

The attractive force applied to each drone to guide it towards its target position is defined as follows:

$$\mathbf{f}_i^{\text{att}} = \alpha \, (\mathbf{t}_i - \mathbf{p}_i) \tag{4}$$

where

- $\mathbf{f}_i^{\text{att}}$ is the attractive force vector for drone i,
- $\mathbf{t}_i$ is the target position of drone i,
- $\mathbf{p}_i$ is the current position of drone i,
- α is a scalar factor ($0 < \alpha \leq 1$), called the *step fraction*, controlling the speed of movement.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table IV. *Swarm formation and flight controller parameters.*

Parameter	Nominal value	Tested range
k_{rep}	0.5	0.1–1.0
r_{rep}	0.25 m	0.1–0.5 m
η_{rep}	0.9	0.7–1.0
α	0.3	0.1–0.5

The critical parameters governing the attractive and repulsive forces – k_{rep}, r_{rep}, η_{rep}, and α – were experimentally tuned to ensure optimised formation accuracy and collision avoidance. The final nominal and tested values, derived from the calibration and validated in the CrazyFlyt simulator, are presented in Table IV.

At each iteration of the swarm control algorithm, the total force acting on each drone is the sum of the attractive and repulsive forces:

$$\mathbf{f}_i = \mathbf{f}_i^{\text{att}} + \mathbf{f}_i^{\text{rep}} \tag{5}$$

The motion of each drone is then autonomously updated through a position-based control model that employs a simple proportional control approach for optimising drone-to-position mappings, enhancing computational efficiency, and ensuring precision. By integrating this control strategy, the system calculates and communicates target setpoints for each drone, facilitating synchronised movement towards predetermined formation coordinates. Formation matrices act as strategic blueprints, guiding drones' arrangement based on recognised gesture shapes. These matrices dynamically adjust in response to input from gesture recognition systems, enabling the swarm to achieve desired formations smoothly and accurately. This integrated approach ensures not only convergence to target formations but also avoidance of inter-agent collisions, while dynamically adapting to gesture-based commands for seamless human–swarm interaction.

The gesture-based drone swarm interaction system, as illustrated in Figure 3, provides a comprehensive overview of the proposed system architecture, which is built on the assumption of homogeneous dynamics and local communication for neighbour detection and collision avoidance within the environment. To ensure stable and synchronised flight, the architecture adheres to strict constraints – notably minimum inter-drone safety distances, maximum step fractions, and limited deployment computational resources. These limitations necessitate the use of optimised, low-latency architectures like BiGRU. The first process in the system begins with the hand gesture acquisition phase, which captures and interprets the physical movements of the user's hands to determine their control intentions while contending with external non-parametric uncertainties, such as variations in lighting and gesture recognition noise arising from user variability. Subsequently, the hand gesture recognition module is employed, integrating advanced computer vision techniques facilitated by MediaPipe for its robust hand-tracking capabilities and BiGRU neural networks for their robust gesture classification. These phases play a crucial role in detecting and understanding the specific actions performed by the user, ensuring that the system can reliably interpret the intended commands. Once a gesture is successfully recognised and classified, it is then mapped to specific control commands through the Gesture-To-Swarm Command Mapping module. This module adapts its mapping logic based on the current operational mode: in Gesture to Formation Command mode, gestures are interpreted as formation patterns, while in Gesture to Flying Command Mode, they are translated into flight commands such as movement directions or altitude adjustments. Following this, the recognised command is processed by the Swarm Formation and Flying Control module, which is specifically designed to mitigate internal parametric uncertainties, most notably drone sensor noise and the presence of faulty robots. By modelling sensor inaccuracies as Gaussian additive noise and accounting for potential drone motor failures, the module employs an adaptive swarm controller that dynamically redistributes attractive and repulsive forces.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

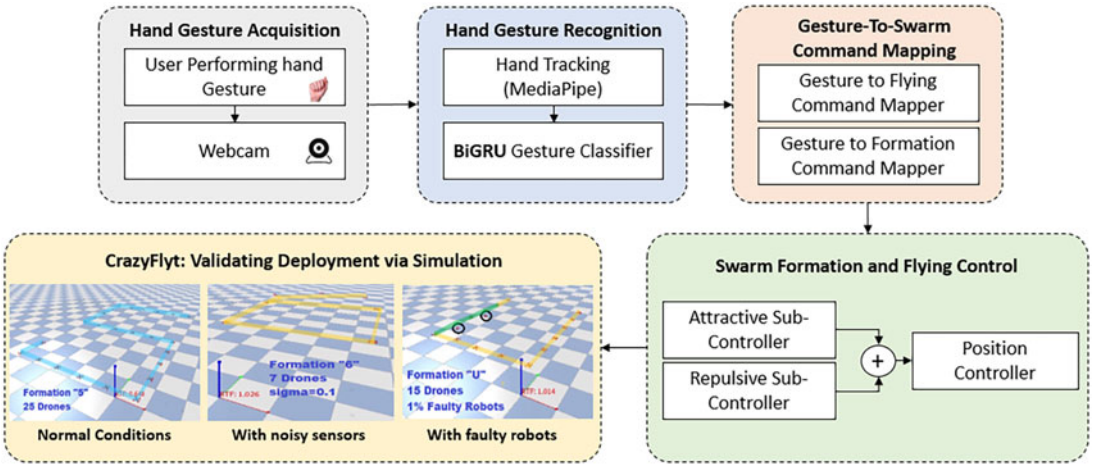


Figure 3. *System-level overview of the proposed gesture-based drone–swarm interaction.*

Validation deployments are performed using the CrazyFlyt simulator – A software library for controlling physical CrazyFlie 2.x drones, featuring integrated simulation capabilities through PyFlyt [35]. The seamless integration of these modules enables the swarm to dynamically adjust its behaviour and adapt to real-time visual cues.

5. Results and discussion

5.1. Evaluation metrics

To comprehensively assess the performance of swarm formation, it is essential to consider multiple evaluation metrics that capture different behavioural aspects. These metrics form a comprehensive framework for evaluating the accuracy, efficiency, consistency, and safety of swarm control strategies.

5.1.1. Average accomplishment time

It represents the mean duration required for the swarm to complete its formation in response to a gesture, measured across n independent simulations. This value is computed as:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{6}$$

5.1.2. The standard deviation (SD) and the coefficient of variation (CV)

They measure the consistency and reliability of formation in multiple trials or gestures [36, 37]. Low values in SD and CV suggest stable and repeatable behaviour. SD and CV are calculated as follows:

$$SD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{7}$$

$$CV = \frac{SD}{\bar{x}} \tag{8}$$

where

- x_i: x is the accomplishment time for the i-th trial
- $\bar{x}$: the mean accomplishment time over n trials
- n: number of trials

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

5.1.3. The Unified Mean Cumulative Error (UMCE)

It reflects the overall accuracy and stability of drone trajectories over time, accounting for deviations from both the target and final positions. When combined with time accomplishment, which indicates how efficiently formation is achieved, UMCE provides a balance between spatial precision and temporal performance.

The UMCE is a useful metric to evaluate swarm formation. Measures how far each drone is from its target and its final position at each step, which helps assess both accuracy and stability. Since the values are averaged over all drones and time steps, the metric is not affected by the number of drones or the duration of the formation. UMCE also highlights slow or unstable behaviour, making it suitable for comparing different control methods or swarm sizes.

$$\text{UMCE} = \frac{1}{2NI} \sum_{t=1}^{I} \sum_{i=1}^{N} \left(\left\| (\mathbf{p}_{i,t} - \mathbf{g}_i) - (\mathbf{p}_{i,t} - \mathbf{p}_{i,I}) \right\| \right) \tag{9}$$

where

- N is the number of drones,
- I is the number of iterations (timesteps),
- $\mathbf{p}_{i,t}$ is the position of drone i at iteration t,
- $\mathbf{g}_i$ is the target (goal) position of drone i,
- $\mathbf{p}_{i,I}$ is the final position of drone i at the last iteration,
- $\|\cdot\|$ denotes the Euclidean distance.

5.1.4. The predicted collision rate

It offers a safety measure by estimating the likelihood of drones approaching dangerously close during formation. In swarm robotics, evaluating and minimising collisions among agents is vital to ensuring both the safety and efficiency of collective behaviour. The pairwise collision approach offers a foundational metric considering the number of agent pairs that enter a critical proximity threshold (collision radius) across multiple simulations, representing all distinct agent pairs that could potentially collide regardless of spatial or temporal constraints. This provides an estimate of the collision risk in the absence of avoidance strategies and is used to evaluate the effectiveness of repulsive control laws. To assess performance over multiple experimental trials, the collision rate can be defined as

$$\text{Predicted Collision Rate} = \frac{\sum_{i=1}^{S} C_i}{S \times \binom{n}{2}} = \frac{2 \sum_{i=1}^{S} C_i}{S \times n(n-1)} \tag{10}$$

where C_i is the number of actual collisions in simulation i, and S is the total number of simulations, and n is the total number of agents in the swarm. This normalised measure allows comparison across different swarm sizes and control strategies, providing a robust indicator of interaction safety. Brambilla et al. [1] emphasise the importance of metrics for designing and validating swarm behaviours, advocating for quantitative assessments of scalability and coordination. Olfati-Saber's framework [38] for multi-agent systems achieves collision avoidance through decentralised control based on inter-agent distances. Together, these works provide a foundation for using pairwise collision metrics in swarm simulations and real-world applications.

5.2. Results of swarm formation control with evaluation metrics

In this study, we highlight the performance of various drone formations with different sizes 7-15-25 drones under three cases: the normal conditions, the effect of noises ($\sigma = 0.1, 0.3$, and 0.5) and the effect of faulty robots (1% and 2%). Using three measures of performance: mean time of accomplishment, SD, and CV, in addition to collision rate analysis. The result is obtained after five simulations per formation

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

with randomly generated initial positions for formations (“A”, “B”, “C”, “H”, “I”, “L”, “O”, “R”, “U”, “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8” and “9”) and predefined start position in formation for the control commands including “up”, “down”, “left”, “right”, “go”, “back”, “takeoff”, “stop”. Applying a repulsive force-based collision avoidance technique. This forecasting technique is adopted to adjust the target path should there be any potential collision in advance, and this contributes substantially to the assessment of each formation’s performance metric.

5.2.1. Results with normal conditions

Based on Table V, which includes the Normal column, and Figure 6(a) (first row) representing the average accomplishment time for each formation, along with Table VI (column Normal) and Figure 7(a) (first column) showing the unified mean error calculation of positions, as well as Table VII (column Normal) representing the SD and CV values, and Figure 8(a) (first column) depicting the predicted collision rate (all in normal conditions) for the different swarm sizes (7-15-25 drones), the analysis demonstrates that:

Under normal conditions, system performance for 7 drones is stable across all formations with minimal variability. Average range accomplishment times narrowly between 17.54 and 17.60 s, indicating that the choice of formation does not significantly impact the accomplishment time under normal conditions. Formations “2” and “6” exhibit the minimum average accomplishment time and CV, indicating high efficiency and consistency. Formation “1” remains the least efficient. Formation “C” shows the highest CV and SD, suggesting some variability in performance related to the fact that the simulations start from random initial positions. Formation “1” exhibits a higher collision rate, suggesting a high possibility of collision, with about 5.71%. The mean cumulative errors are minimal for most formations, indicating that the collision avoidance system is effective in maintaining precise formations when there are no external disturbances or faults. This suggests that the repulsive forces are well-calibrated, allowing drones to maintain their intended trajectories without significant deviations. This is except for formation “2”, where the system consistently deviates from the target position due to the applied repulsive force preventing collisions (approximately 1.9% of potential collisions). In addition to formation “R” with remarkable error variance due to collision prevention with a rate of 4.76%.

For 15 drones, the average time is around 17.21 to 17.76 s, indicating minimal variance in performance. Formation “U” has the lowest average time, meaning the most efficient formation, while formation “B” exhibits the highest average time. Formations “R” and “U” show high CV values, indicating high variability in their accomplishment times relative to their means. Formation “C” has the lowest CV values, indicating no variability, with a significant error value indicating a deviation from the target, followed by formations “7” and “O” with a collision rate of 7.62%, while formation “A” exhibits the lowest error value, indicating efficiency in terms of formation maintenance.

For 25 drones, formation “A” demonstrates consistent completion times of 19.04 s, exhibiting no variability and a high level of precision. In contrast, formation “U” shows poorer overall performance with 19.20 s. This time range shows an increase of 2 s for a larger swarm size, indicating that the complexity of size impacts the time performance and accuracy. Formations “9”, “H”, and “O” express notable cumulative mean errors (UMCE), indicating necessary adjustments in their paths due to the applied repulsive forces to prevent future collisions. These formations experience a collision rate of approximately 1%.

The following Figure 4 shows the trajectories of various formations under normal conditions for different swarm sizes. The first row displays the trajectories for 7 drones, the second row for 15 drones, and the third row for 25 drones.

Trajectory profiles in drone swarms hinge on two main factors: the density of the swarm and whether collision-avoidance protocols are active. Under normal conditions (without noise, no faulty drones), a small swarm size of seven drones (Formation “A” in Figure 4). Drones are spaced so that each one can proceed directly to its assigned target. Paths taken are simply the shortest straight (Euclidean) lines. As long as drones maintain pairwise distances above the safety threshold, the collision avoidance system remains passive; it does not intervene. The system monitors for risks but takes no action unless necessary. Resulting in uninterrupted, straight, and collision-free paths. Even when flight paths cross geometrically,

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table V. *Average formation accomplishment time in seconds for different swarm size 7-15-25 in three cases: normal conditions, effect of sensor noise (with σ 0.1, 0.3 and 0.5) and effect of 1% and 2% of faulty robots.*

		Noise			Faulty robots	
Formation	**Normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**
			7 drones			
"A"	17.57	17.62	17.73	17.56	27.29	27.32
"B"	17.56	17.62	17.56	17.55	27.31	27.33
"C"	17.58	17.64	17.56	17.55	27.28	27.34
"H"	17.57	17.64	17.62	17.56	27.28	27.30
"I"	17.60	17.61	17.55	17.57	27.33	27.29
"L"	17.58	17.57	17.54	17.56	27.33	27.29
"O"	17.57	17.58	17.56	17.55	27.28	27.29
"R"	17.58	17.67	17.55	17.55	27.28	27.29
"U"	17.58	17.44	17.55	17.60	27.29	27.29
"1"	17.56	17.25	17.57	17.47	27.30	27.29
"2"	17.54	17.67	17.56	17.56	27.28	27.40
"3"	17.56	17.68	17.57	17.58	27.29	27.34
"4"	17.56	17.60	17.57	17.58	27.29	27.38
"5"	17.58	17.60	17.56	17.57	27.31	27.39
"6"	17.54	17.60	17.58	17.59	27.33	27.47
"7"	17.56	17.66	17.60	17.57	27.33	27.34
"8"	17.58	17.58	17.58	17.60	27.33	27.28
"9"	17.58	17.58	17.57	17.56	27.33	27.29
			15 drones			
"A"	17.71	17.76	17.70	17.75	27.83	27.49
"B"	17.76	17.71	17.72	17.79	27.59	27.49
"C"	17.67	17.71	17.73	17.73	27.50	27.59
"H"	17.67	17.74	17.77	17.71	27.48	27.58
"I"	17.68	17.71	17.72	17.68	27.50	27.55
"L"	17.68	17.72	17.72	17.70	27.96	27.47
"O"	17.68	17.59	17.60	17.63	27.66	27.50
"R"	17.43	17.72	17.47	17.77	27.52	27.48
"U"	17.21	17.62	17.34	17.33	27.49	27.61
"1"	17.67	17.67	17.68	17.71	27.49	27.76
"2"	17.55	17.67	17.70	17.69	27.49	27.61
"3"	17.68	17.74	17.68	17.71	27.48	27.61
"4"	17.69	17.74	17.72	17.68	27.48	27.53
"5"	17.67	17.68	17.69	17.71	27.48	27.53
"6"	17.68	17.70	17.71	17.73	27.50	27.51
"7"	17.67	17.69	17.67	17.70	27.83	27.55
"8"	17.69	17.74	17.81	17.73	27.50	27.59
"9"	17.69	17.74	17.73	17.72	27.49	27.61
			25 drones			
"A"	19.04	19.40	19.40	19.54	31.37	29.27
"B"	19.04	19.56	19.56	19.65	31.71	30.66

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table V. *Continued*

		Noise			Faulty robots	
Formation	**Normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**
"C"	19.05	19.54	19.55	19.57	31.59	29.99
"H"	19.07	19.40	19.42	19.47	31.38	29.71
"I"	19.13	19.52	19.50	19.50	31.57	29.84
"L"	19.13	19.76	19.33	19.29	31.41	29.93
"O"	19.19	19.41	19.42	19.52	28.86	29.92
"R"	19.12	19.31	19.45	19.53	30.79	29.91
"U"	19.20	19.49	19.56	19.64	30.74	29.88
"1"	19.13	19.36	19.41	19.41	30.59	29.93
"2"	19.14	19.36	19.35	19.36	30.19	29.76
"3"	19.12	19.53	19.54	19.61	30.62	30.15
"4"	19.11	19.50	19.49	19.44	30.70	30.16
"5"	19.14	19.35	19.36	19.38	30.43	30.15
"6"	19.16	19.40	19.42	19.52	30.39	30.26
"7"	19.16	19.18	19.20	19.27	30.15	29.96
"8"	19.06	19.24	19.41	19.47	30.35	29.92
"9"	19.10	19.16	19.31	19.32	29.86	30.14

drones reach intersection points at different times, so no actual collisions occur. The system avoids unnecessary deviations, so no energy or time is wasted manoeuvring around "phantom" obstacles. The controller employs linear interpolation, which naturally produces straight-line trajectories from start to goal.

When the swarm consists of a larger number of drones, like 25 in Formation "3" (see Figure 4), the airspace becomes crowded. Increased density leads to a higher chance of overlapping flight paths and potential conflicts between drones. The collision avoidance system activates in these conditions, applies repulsive forces between drones and causes local deviations in some trajectories to prevent collisions. Maintains overall safety and ensures that drones eventually reach their designated positions. These deviations are clear evidence of the collision avoidance system's effectiveness; no collisions occur throughout the process. The final formation is successfully achieved. In less crowded situations, drones follow near-straight, optimal paths. The collision avoidance system remains mostly inactive here, since path conflicts are unlikely.

5.2.2. Effects of sensor noises

Based on the data presented in Table V, which includes the "Noise" column, and the average accomplishment time for each formation illustrated in Figure 6(a) (second row), we can analyse the results. Additionally, Table VI (column "Noise") and Figure 7(a) (second column) provide insights into the unified mean error calculation of positions. Furthermore, Table VII (column "Noise") outlines the SD and CV values, while Figure 8(a) (second column) shows the predicted collision rates under normal conditions for different swarm sizes (7, 15, and 25 drones). This comprehensive analysis leads us to the following conclusions:

The accomplishment times for 7 drones remain relatively stable across varying noise levels, indicating resilience to noise between 17.25 and 17.73 s. However, slight increases in time can be observed with higher noise levels compared to normal conditions, not exceeding 1%. Formation "1" achieves the lowest average accomplishment times under varying noise levels, indicating robustness to noise. The predicted collision rates increase when noise levels increase, particularly for scalable swarms for some formations, suggesting that significant noise impacts the effectiveness of the collision avoidance system. Formations

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table VI. *Formation Unified Mean Cumulative Error (UMCE) calculating for different swarm size 7-15-25.*

Formation	Normal	Noise			Faulty robots	
		$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	1%	2%
			7 drones			
"A"	6.81E-07	1.59E-01	4.77E-01	7.95E-01	4.24E-07	6.19E-07
"B"	1.43E-07	1.55E-01	4.64E-01	7.73E-01	2.86E-07	2.57E-07
"C"	7.44E-07	2.13E-01	6.40E-01	1.07E+00	7.45E-07	8.07E-07
"H"	5.91E-07	1.33E-01	4.00E-01	6.67E-01	3.38E-07	4.12E-07
"I"	2.14E-07	1.22E-01	3.66E-01	6.09E-01	2.57E-07	2.00E-07
"L"	3.57E-07	1.24E-01	3.72E-01	6.20E-01	4.00E-07	2.86E-07
"O"	2.86E-07	1.19E-01	3.56E-01	5.93E-01	2.00E-07	3.26E-07
"R"	5.02E-01	2.44E+00	3.67E-01	6.11E-01	1.83E-07	2.86E-07
"U"	2.38E-07	1.98E-01	5.94E-01	9.90E-01	2.69E-07	3.26E-07
"1"	1.43E-07	1.49E-01	4.46E-01	7.43E-01	2.00E-07	2.69E-07
"2"	2.60E+00	1.37E-01	4.12E-01	6.86E-01	3.14E-07	3.55E-07
"3"	3.55E-07	1.76E-01	5.27E-01	8.78E-01	3.14E-07	1.71E-07
"4"	2.00E-07	1.09E-01	3.28E-01	5.47E-01	2.29E-07	3.43E-07
"5"	2.24E-07	1.86E-01	5.58E-01	9.29E-01	2.12E-07	2.40E-07
"6"	3.71E-07	1.58E-01	4.75E-01	7.92E-01	2.69E-07	2.57E-07
"7"	4.29E-07	1.51E-01	4.53E-01	7.54E-01	6.00E-07	4.98E-07
"8"	2.29E-07	1.35E-01	9.70E-01	6.73E-01	2.86E-07	3.09E-07
"9"	2.57E-07	1.79E-01	5.37E-01	8.95E-01	2.57E-07	2.12E-07
			15 drones			
"A"	1.60E-07	1.71E-01	5.14E-01	8.57E-01	1.33E-07	2.06E-07
"B"	1.79E-07	1.43E-01	4.30E-01	7.16E-01	1.79E-07	1.73E-07
"C"	7.85E-07	1.65E-01	4.94E-01	8.23E-01	8.75E-07	7.13E-07
"H"	3.84E-07	1.52E-01	4.57E-01	7.62E-01	2.70E-07	4.09E-07
"I"	2.93E-07	1.83E-01	5.50E-01	9.17E-01	3.73E-07	3.20E-07
"L"	3.73E-07	1.49E-01	4.46E-01	1.19E+00	2.00E-07	3.20E-07
"O"	4.18E-07	1.63E-01	4.88E-01	8.13E-01	4.26E-07	4.28E-07
"R"	2.72E-07	8.14E-01	9.55E-01	6.66E-01	9.33E-08	9.33E-08
"U"	2.35E-07	1.07E+00	9.13E-01	8.05E-01	6.67E-08	8.55E-08
"1"	2.93E-07	1.50E-01	4.50E-01	1.41E+00	3.23E-07	2.59E-07
"2"	3.26E-07	1.52E-01	4.56E-01	7.60E-01	2.67E-07	1.33E-07
"3"	1.73E-07	1.56E-01	4.69E-01	7.81E-01	1.20E-07	2.59E-07
"4"	3.04E-07	1.62E-01	4.85E-01	8.09E-01	1.47E-07	2.27E-07
"5"	2.57E-07	1.74E-01	5.23E-01	8.72E-01	2.99E-07	3.12E-07
"6"	2.06E-07	1.35E-01	4.04E-01	6.73E-01	1.66E-07	2.53E-07
"7"	5.42E-07	1.19E-01	3.58E-01	5.97E-01	3.39E-07	6.09E-07
"8"	1.73E-07	1.60E-01	4.81E-01	8.01E-01	1.47E-07	1.98E-07
"9"	1.87E-07	1.80E-01	5.41E-01	9.02E-01	2.99E-07	1.47E-07
			25 drones			
"A"	8.80E-08	1.52E-01	4.57E-01	7.62E-01	8.33E-08	4.00E-08
"B"	9.66E-08	1.59E-01	4.77E-01	7.96E-01	8.33E-08	9.60E-08

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table VI. *Continued.*

Formation	Normal	Noise			Faulty robots	
		$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	1%	2%
"C"	1.07E-07	1.51E-01	4.53E-01	7.55E-01	7.20E-08	8.00E-08
"H"	1.68E+00	1.64E-01	4.92E-01	8.20E-01	1.25E-07	9.46E-08
"I"	7.20E-08	1.59E-01	4.78E-01	7.97E-01	9.13E-08	8.33E-08
"L"	1.03E-07	1.70E-01	5.11E-01	8.52E-01	1.17E-07	9.46E-08
"O"	1.00E+00	1.55E-01	4.65E-01	7.74E-01	3.18E-07	3.33E-07
"R"	1.52E-07	1.46E-01	4.38E-01	7.30E-01	7.20E-08	8.80E-08
"U"	8.33E-08	1.69E-01	5.07E-01	8.46E-01	1.40E-07	8.66E-08
"1"	8.18E-01	1.51E-01	4.52E-01	7.54E-01	1.06E-07	1.01E-07
"2"	8.99E-08	1.58E-01	4.73E-01	7.88E-01	8.00E-08	7.06E-08
"3"	5.46E-08	1.60E-01	4.81E-01	8.02E-01	9.13E-08	8.19E-08
"4"	5.60E-08	1.22E-01	3.67E-01	6.12E-01	9.93E-08	8.19E-08
"5"	9.13E-08	1.79E-01	5.36E-01	8.93E-01	8.33E-08	1.17E-07
"6"	1.35E-07	1.57E-01	4.71E-01	7.85E-01	1.11E-07	8.80E-08
"7"	5.13E-08	1.52E-01	4.55E-01	7.58E-01	1.71E-07	8.19E-08
"8"	7.86E-08	1.80E-01	5.41E-01	9.02E-01	1.35E-07	7.86E-08
"9"	1.26E+00	1.48E-01	4.45E-01	7.42E-01	1.23E-07	7.06E-08

such as "R" ($\sigma = 0.1$), "U", "3" and "C" ($\sigma = 0.3$ and 0.5) exibit higher collision rates under increased noise, with a significant increase in cumulative mean error between 5.71% and 7.62%, especially for formations "R" and "3", indicating that while the collision avoidance system can handle some level of uncertainty, higher noise levels introduce more variability in drone positions.

The results for 15 drones indicate stable performance. The time performance is still approximately in the same cluster, with minimum variation between 17.33 and 17.81 s, remaining stable, representing less than a 1% variation relative to the normal case. Whereas the formation "U" shows a time performance. "O" and "R" express the highest SD and CV values overall, meaning some variances according to the two measures and a cumulative mean error in the range of 0.9–1.5. The collision rates indicate that formations "4" and "1" have significant potential collisions, ranging between 0.76% and 2.48%. These formations also have an adjustment of trajectories to prevent collisions, which is reflected in UMCE values. Les formations "1", "L" ($\sigma = 0.5$), "U" et "R" ($\sigma = 0.1$ et 0.3) present high error measurements, indicating that these formations, at different noise levels, require trajectory adjustments to prevent collisions, which is confirmed by the high collision rate, particularly for formation "1" with a rate of 0.76%.

However, the time performance for 25 drones is higher with an additional 2 s compared to 7 and 15 drones, related to the additional swarm and the complexity of coordination for a scalable architecture. Formations "L" and "B" express the low performances over the different levels of Gaussian noise. Formations "7" and "9" are well performed. Formations "8", "5" and "U" show low accuracy in terms of trajectories due to a high cumulative mean error calculation above 0.1 for the three levels of noise, while formation "4" represents the optimum trajectories with the lowest error value.

Figure 5 illustrates the trajectories of various formations while introducing Gaussian noise at levels of 0.1, 0.3, and 0.5 for different swarm sizes. The first row displays the trajectories for 7 drones, with levels 0.1, 0.3, and 0.5 represented in columns 1, 2, and 3, respectively. The second row shows the trajectories for 15 drones, and the third row presents the trajectories for 25 drones.

The introduction of noise stemming from sensor inaccuracies, actuator inconsistencies, or communication glitches causes drones to stray from their ideal straight-line trajectories. Rather than following flawless routes, drones display irregular, less predictable movement patterns under noisy conditions. While the collision avoidance system remains effective and successfully prevents collisions, the cumulative effect of noise introduces residual positioning errors. The magnitude of these positioning errors

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table VII. *Formation standard deviation (SD) and coefficient of variation (CV) calculating for different swarm size 7-15-25 in three cases: normal conditions, effect of sensor noise (with σ 0.1, 0.3 and 0.5) and effect of 1% and 2% of faulty robots.*

		SD noise			SD faulty robots			CV noise			CV faulty robots	
Formation	**SD normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**	**CV normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**
					7 drones							
"A"	0.03	0.03	0.10	0.02	0.01	0.00	0.0020	0.0020	0.0056	0.0009	0.0005	0.0001
"B"	0.02	0.13	0.02	0.02	0.04	0.01	0.0014	0.0073	0.0013	0.0011	0.0014	0.0003
"C"	0.06	0.12	0.02	0.01	0.01	0.01	0.0032	0.0069	0.0010	0.0004	0.0004	0.0004
"H"	0.02	0.07	0.06	0.02	0.01	0.01	0.0010	0.0042	0.0036	0.0009	0.0002	0.0004
"I"	0.02	0.08	0.02	0.03	0.01	0.01	0.0014	0.0044	0.0010	0.0015	0.0003	0.0002
"L"	0.02	0.01	0.00	0.02	0.02	0.01	0.0013	0.0008	0.0002	0.0011	0.0006	0.0003
"O"	0.04	0.04	0.02	0.02	0.01	0.01	0.0023	0.0025	0.0014	0.0010	0.0003	0.0003
"R"	0.02	0.06	0.01	0.02	0.01	0.01	0.0012	0.0034	0.0003	0.0012	0.0004	0.0003
"U"	0.03	0.22	0.01	0.04	0.01	0.00	0.0018	0.0128	0.0005	0.0024	0.0003	0.0001
"1"	0.01	0.26	0.04	0.18	0.02	0.02	0.0007	0.0153	0.0022	0.0100	0.0008	0.0006
"2"	0.00	0.10	0.02	0.02	0.01	0.06	0.0003	0.0059	0.0010	0.0012	0.0002	0.0023
"3"	0.03	0.10	0.03	0.02	0.01	0.02	0.0019	0.0056	0.0018	0.0013	0.0004	0.0008
"4"	0.04	0.04	0.03	0.03	0.01	0.06	0.0021	0.0021	0.0016	0.0019	0.0003	0.0022
"5"	0.04	0.04	0.01	0.03	0.02	0.07	0.0022	0.0022	0.0008	0.0017	0.0006	0.0026
"6"	0.01	0.06	0.03	0.03	0.00	0.09	0.0005	0.0032	0.0015	0.0016	0.0001	0.0032
"7"	0.03	0.08	0.03	0.03	0.01	0.02	0.0019	0.0044	0.0015	0.0015	0.0003	0.0007
"8"	0.06	0.02	0.03	0.02	0.01	0.00	0.0033	0.0011	0.0015	0.0012	0.0003	0.0002
"9"	0.05	0.04	0.04	0.02	0.06	0.01	0.0030	0.0022	0.0021	0.0010	0.0021	0.0003
					15 drones							
"A"	0.02	0.08	0.04	0.02	0.29	0.01	0.0010	0.0044	0.0023	0.0013	0.0105	0.0004
"B"	0.03	0.04	0.02	0.10	0.02	0.02	0.0016	0.0022	0.0014	0.0054	0.0008	0.0006
"C"	0.00	0.04	0.03	0.04	0.04	0.02	0.0001	0.0020	0.0014	0.0021	0.0014	0.0007
"H"	0.01	0.06	0.11	0.03	0.00	0.01	0.0005	0.0032	0.0064	0.0016	0.0002	0.0003
"I"	0.01	0.03	0.04	0.01	0.05	0.06	0.0007	0.0018	0.0021	0.0008	0.0018	0.0023
"L"	0.01	0.04	0.03	0.02	0.22	0.00	0.0007	0.0022	0.0017	0.0009	0.0078	0.0002
"O"	0.01	0.25	0.24	0.29	0.24	0.01	0.0004	0.0139	0.0135	0.0163	0.0085	0.0004
"R"	0.29	0.04	0.29	0.02	0.08	0.01	0.0164	0.0025	0.0168	0.0010	0.0028	0.0003

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table VII. Continued.

		SD noise			SD faulty robots			CV noise			CV faulty robots	
Formation	**SD normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**	**CV normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**
"U"	0.24	0.15	0.25	0.22	0.02	0.12	0.0138	0.0086	0.0143	0.0128	0.0008	0.0043
"1"	0.01	0.01	0.02	0.05	0.01	0.05	0.0003	0.0008	0.0010	0.0029	0.0005	0.0019
"2"	0.24	0.01	0.04	0.01	0.02	0.03	0.0134	0.0003	0.0020	0.0008	0.0007	0.0010
"3"	0.01	0.01	0.01	0.02	0.01	0.01	0.0008	0.0007	0.0004	0.0013	0.0003	0.0005
"4"	0.01	0.04	0.04	0.01	0.01	0.01	0.0006	0.0022	0.0020	0.0006	0.0002	0.0005
"5"	0.00	0.01	0.04	0.03	0.02	0.03	0.0003	0.0008	0.0022	0.0019	0.0006	0.0010
"6"	0.01	0.06	0.06	0.05	0.01	0.03	0.0005	0.0036	0.0036	0.0028	0.0004	0.0010
"7"	0.01	0.03	0.01	0.03	0.28	0.04	0.0004	0.0016	0.0003	0.0017	0.0101	0.0015
"8"	0.02	0.01	0.10	0.05	0.01	0.00	0.0009	0.0005	0.0056	0.0027	0.0005	0.0002
"9"	0.01	0.02	0.04	0.04	0.00	0.04	0.0008	0.0009	0.0020	0.0021	0.0002	0.0014
					25 drones							
"A"	0.02	0.28	0.29	0.33	0.27	0.44	0.0010	0.0142	0.0150	0,0167	0.0092	0.0141
"B"	0.03	0.31	0.22	0.01	0.19	0.26	0.0013	0.0161	0.0111	0,0005	0.0061	0.0083
"C"	0.04	0.28	0.22	0.33	0.37	0.28	0.0019	0.0141	0.0111	0,0168	0.0125	0.0089
"H"	0.03	0.20	0.33	0.18	0.32	0.41	0.0017	0.0104	0.0169	0,0091	0.0108	0.0131
"I"	0.08	0.23	0.18	0.18	0.45	0.29	0.0042	0.0116	0.0094	0,0093	0.0151	0.0093
"L"	0.10	0.26	0.31	0.30	0.60	0.13	0.0054	0.0131	0.0159	0,0155	0.0201	0.0042
"O"	0.09	0.17	0.20	0.22	0.59	0.20	0.0049	0.0086	0.0101	0,0114	0.0196	0.0070
"R"	0.10	0.28	0.27	0.29	0.30	0.07	0.0052	0.0143	0.0140	0,0149	0.0100	0.0023
"U"	0.05	0.25	0.25	0.22	0.06	0.12	0.0024	0.0129	0.0129	0,0111	0.0020	0.0040
"1"	0.09	0.23	0.25	0.22	0.47	0.23	0.0046	0.0118	0.0128	0,0113	0.0157	0.0076
"2"	0.09	0.27	0.30	0.22	0.21	0.04	0.0046	0.0137	0.0156	0,0112	0.0072	0.0014
"3"	0.10	0.28	0.33	0.30	0.02	0.23	0.0054	0.0146	0.0169	0,0152	0.0007	0.0075
"4"	0.10	0.34	0.27	0.25	0.02	0.21	0.0055	0.0177	0.0138	0,0126	0.0007	0.0067
"5"	0.13	0.24	0.34	0.33	0.01	0.21	0.0068	0.0123	0.0173	0,0171	0.0004	0.0071
"6"	0.12	0.30	0.29	0.32	0.10	0.29	0.0065	0.0154	0.0148	0,0163	0.0033	0.0096
"7"	0.10	0.25	0.22	0.24	0.12	0.02	0.0054	0.0129	0.0116	0,0123	0.0038	0.0007
"8"	0.05	0.20	0.30	0.22	0.29	0.26	0.0028	0.0103	0.0155	0,0111	0.0096	0.0087
"9"	0.07	0.25	0.33	0.30	0.37	0.10	0.0034	0.0131	0.0173	0,0156	0.0123	0.0034

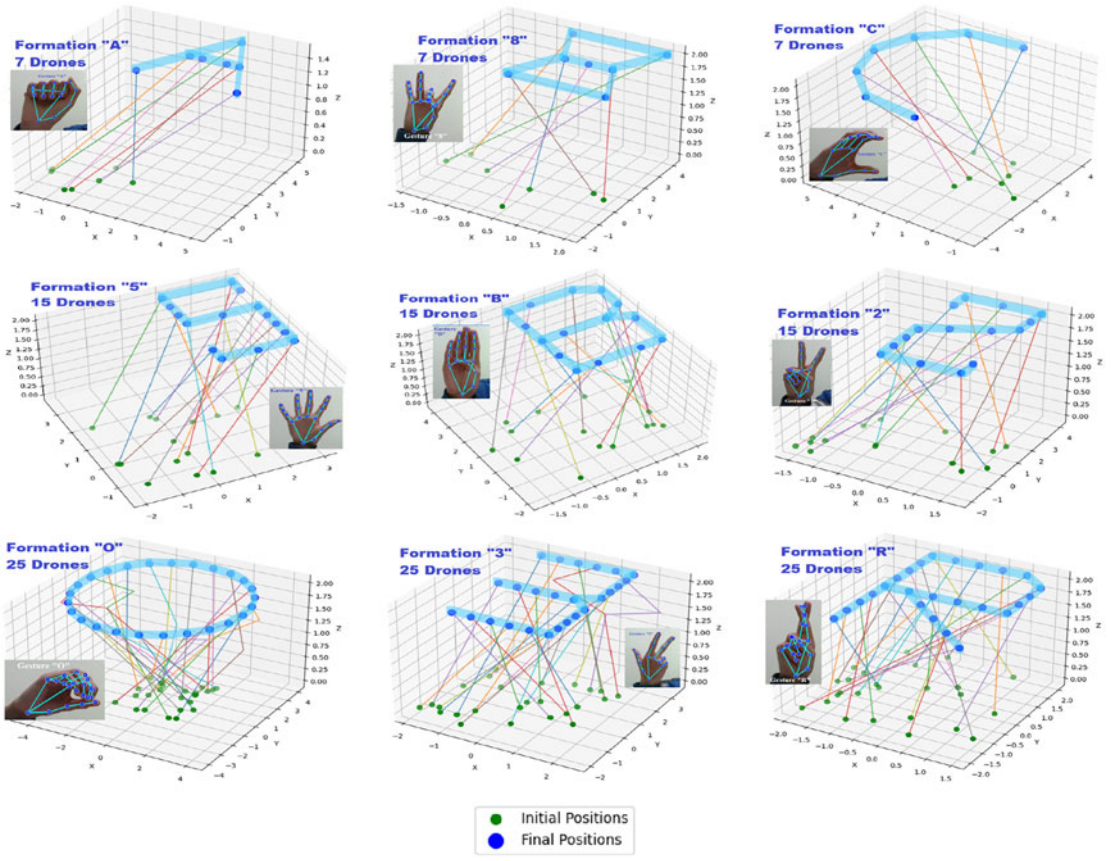


Figure 4. *Trajectories of swarm formations composed of 7, 15, and 25 drones, generated from recognised hand gestures. The figure illustrates the evolution of drone positions from random initial states to the final target geometric configurations, highlighting the convergence behaviour of the proposed gesture-based control approach.*

is directly linked to the noise level present in the system. Some drones achieve near-perfect alignment with their intended targets, while others demonstrate more noticeable displacement. The resulting swarm formation is close to, but not exactly, the desired configuration. This outcome realistically reflects operating scenarios where environmental disturbances are unavoidable. While the system robustly maintains safety and prevents collisions, perfect formation accuracy becomes unattainable in the presence of noise.

5.2.3. Effect of faulty robots

Based on Table V, which includes the "Faulty Robots" column with two cases (1% and 2% of Faulty Robots), and Figure 6(c) (third row) representing the average accomplishment time for each formation, along with Table VI (column "Faulty Robots") and Figure 7(c) (third column) showing the unified mean error calculation of positions, as well as Table VII (column Normal) representing the SD and CV values, and Figure 8(c) (third column) depicting the predicted collision rate (all in normal conditions) for the different swarm sizes (7-15-25 drones), the analysis demonstrates that:

In this study, the drones with faults were simulated to be non-responsive for 5 s in random iterations to replicate real-world faults in drone coordination and communications. The simulation highlights the supreme importance of ensuring drone reliability in formations. Even a very minimal fault rate can produce significant degradation in performance, and thus, robust fault detection and mitigation mechanisms are necessary to maintain efficiency in drone operations.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

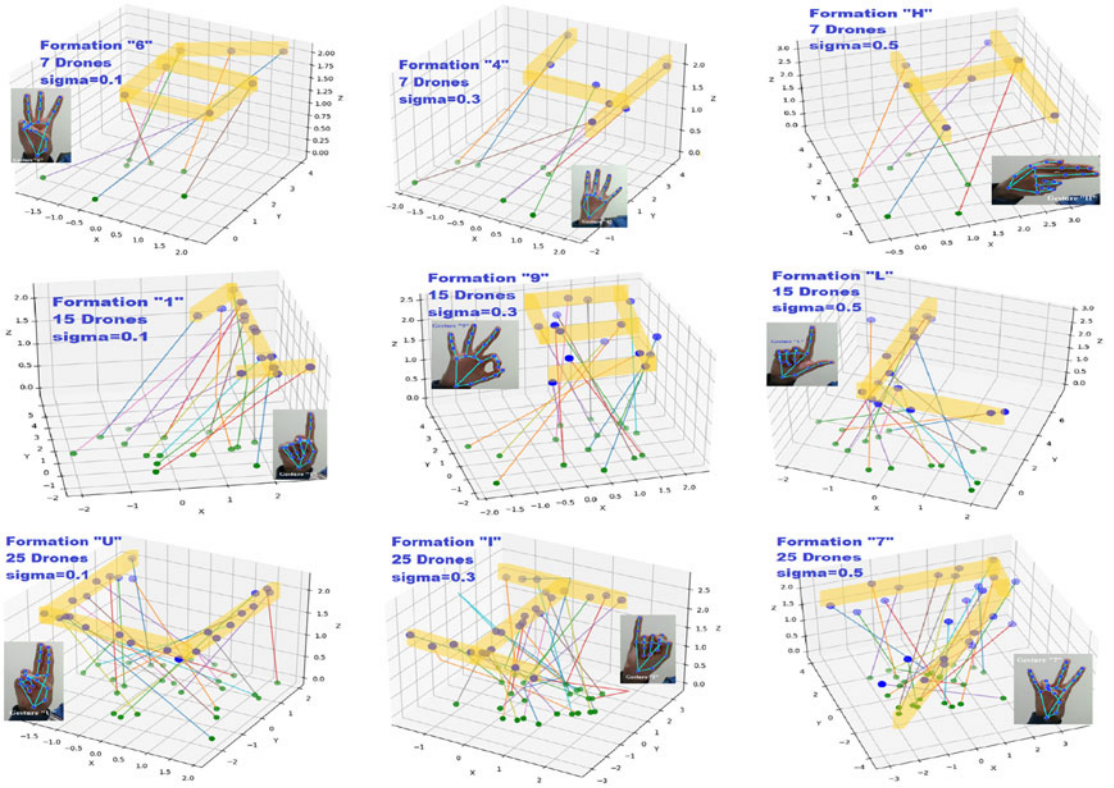


Figure 5. *Trajectories of swarms composed of 7, 15, and 25 drones under recognised hand gesture-based control, evaluated across noise levels ($\sigma = 0.1, 0.3, 0.5$). The results, obtained in simulation, show the impact of increasing noise on formation trajectories and demonstrate the robustness of the proposed approach.*

For seven drones, the average accomplishment time ranges between 27.28 and 27.96 s. When faulty drones are introduced, this corresponds to an increase in accomplishment times of approximately over 50% increase relative to the normal case, compared with the noise levels, which do not exceed 1%. Formations "7" and "L" exhibit reduced performance under faulty conditions (with 1% and 2% faults), indicating a sensitivity to faults. Otherwise, the formation "H" demonstrates efficiency for both faulty scenarios. Formation "R" continues to show higher collision rates of 5.71% and 6.67% for 1 faulty drone and 2 faulty drones, respectively, but it still achieves performance in terms of time with the minimal error values. Formation "C" exhibits a high error value of 7.45E-7 and 8.07E-7 in both faulty scenarios. Formations "H", "O", "5", and "9" are recognised as the best formations without collision for the first faulty scenarios (1 faulty drone). In the case of two faulty drones, formations "B", "H" and "L" perform well, with formation "H" being considered a highly adaptable collision-free formation over all simulated faulty cases. The SD values indicate that all formations have a small spread around the mean of approximately 0.09. The CV values suggest low variability, all under 0.0032, indicating that the time accomplishment is consistent and stable.

For 15 drones, the efficient completion time is 27.28 s for formation "8", where formation "I" is considered the low-performing formation with 27.76 s representing over 50% increase in time compared to the normal case. Formation "C" exhibits a minimal adjustment at the level of trajectories to achieve the proper formation, according to the error values range from 7.13E-7 to 8.75E-7 for both scenarios (cases 2 and 3 faulty drones). Formation "U" expresses the best optimum path with the lowest error values. The collision rate shows that the most formation exposed to collision is formation "O" with a 0.95% rate for the first scenario, while formation "U" has a 1.14% rate for the second scenario. SD and CV values

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

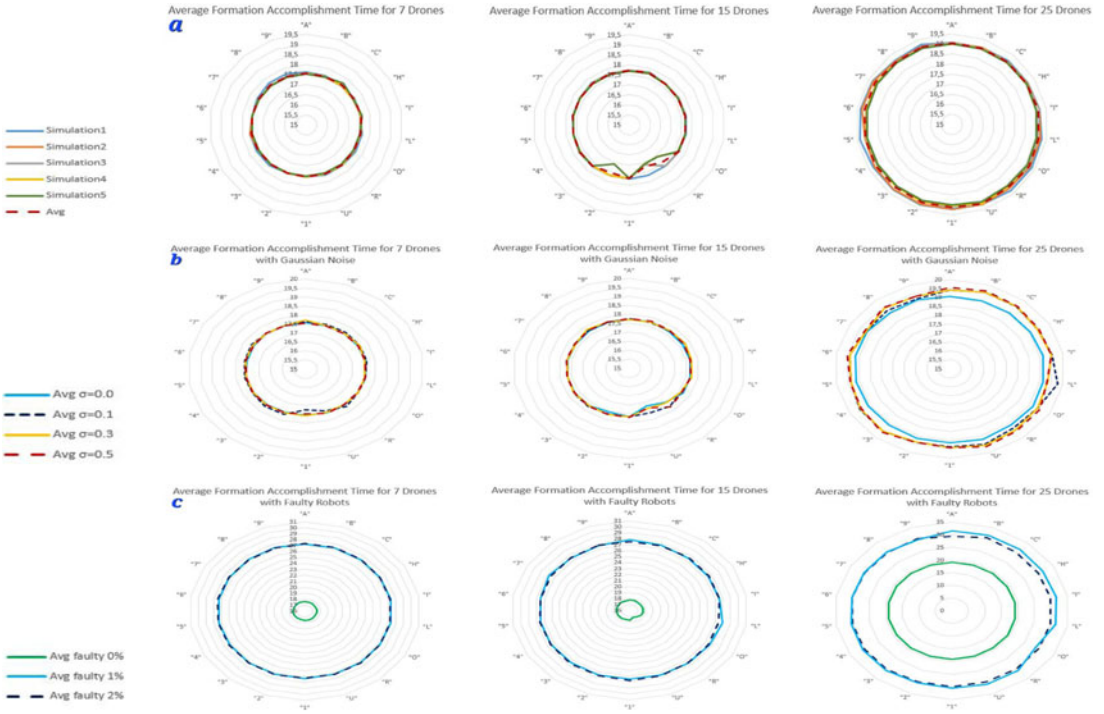


Figure 6. *Average formation accomplishment time (seconds) for swarms of 7, 15, and 25 drones under different formation tasks. The results are evaluated under three scenarios: normal conditions, sensor noise levels ($\sigma = 0.1, 0.3, 0.5$), and faulty robot ratios (1% and 2%). The figure illustrates the impact of uncertainty and faults on the formation time and scalability of the proposed approach.*

show that formation "A" and "U" respectively (for 1% and 2% faulty drones) are the highest values (0.29 and 1.05% for SD and CV, respectively). These values are considered low variability, suggesting the formations are consistent.

By scaling the swarm size to 25 drones, the average completion time is increased. Formation "A" exhibits the longest time to achieve the formation above 31.71 s for three faulty drones. Formation "B" is less efficient, with five faulty drones (30.66 s), with the highest collision rate of 6.07% over all formations. Formations "O" and "A" are considered efficient for two faulty scenarios, respectively. UMCE values indicate that all trajectories are stable, with the most variable for formation "O" for both cases, with an error under 3.3E-7. All formations faced a potential collision for 25 drones with the minimum value of 2.2% rate, suggesting that this architecture is more complex than the others. SD and CV values show high consistency except for formation "L", with an SD of 0.6, suggesting that the formation is faced with collisions and the adjustment of positions to achieve the final form.

As shown in Figure 9, swarms of 7, 15, and 25 drones can form and hold formations reliably under normal, controlled conditions. Increasing the number of drones primarily results in longer completion times. Notably, accuracy remains consistent regardless of swarm size. Introducing Gaussian disturbances (simulating environmental noise) leads to less than 1% increase in completion time and noticeable rise in trajectory deviations and collision probabilities, especially in larger swarms. The system handles minor environmental noise well, but larger swarms reveal vulnerabilities related to scaling. Introducing a faulty drone (paused for 5 s without trajectory updates) results in over 50% increase in completion times. Collision risks rise proportionally with swarm size. The remainder of the swarm must wait for the faulty unit to resume, causing delays and heightening collision probability. The faulty-drone setup is designed to mimic real-world issues such as temporary communication disruptions, connection losses, and synchronisation errors within the swarm.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

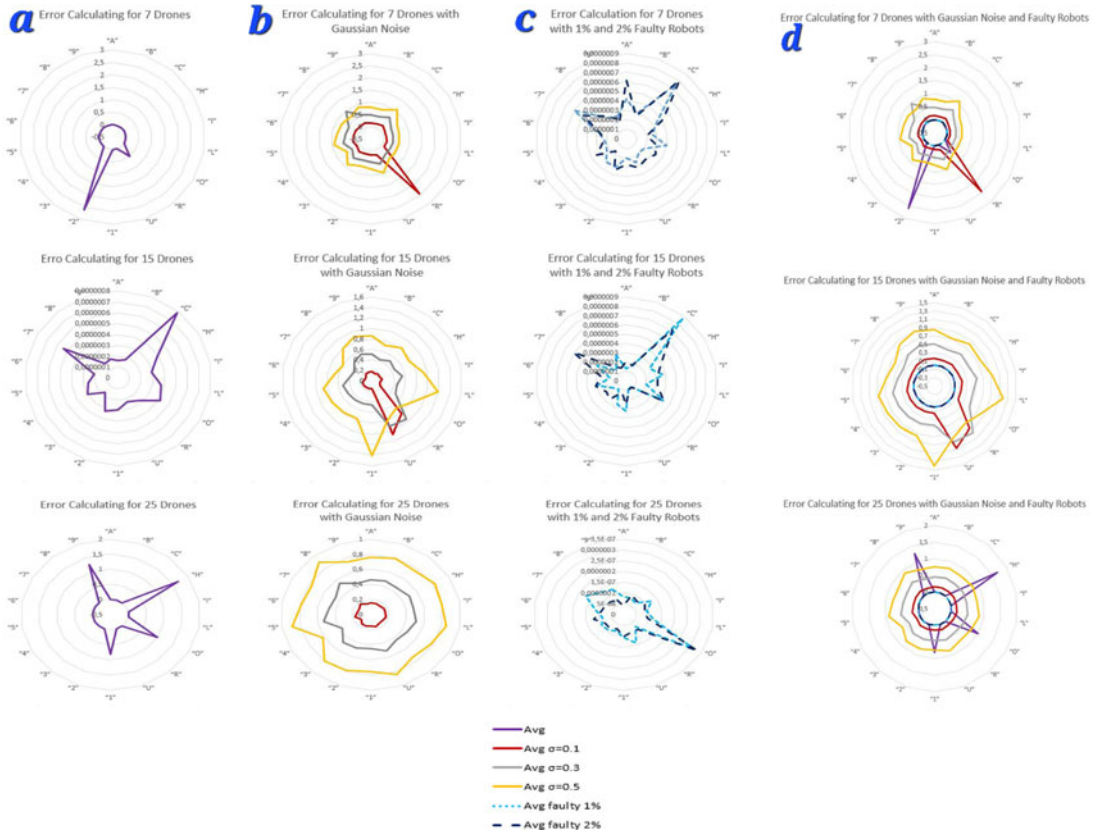


Figure 7. *Average formation error for drone swarms composed of 7, 15, and 25 drones under different formation tasks. The figure analyses formation performance and scalability as the swarm size increases.*

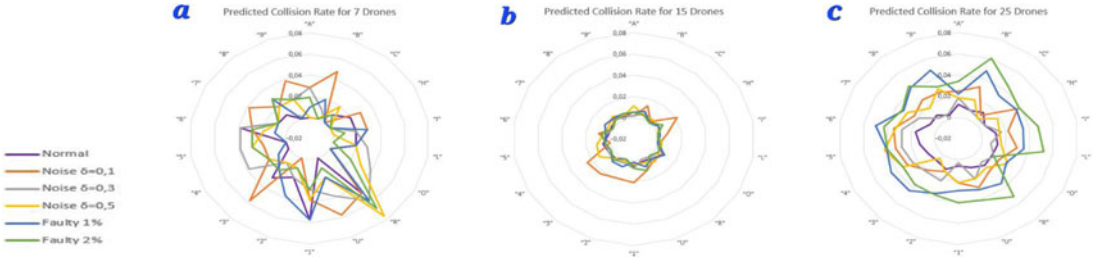


Figure 8. *Collision rate of potential formation collisions for swarms composed of 7, 15, and 25 drones under different formation tasks. The figure analyses collision occurrence and scalability behaviour as swarm size increases.*

The system demonstrates resilience to minor noise disturbances. Scalability is conditional on the ability to detect and manage temporary drone faults effectively. Incorporating fault-tolerant strategies is essential for maintaining reliable swarm coordination and formation efficiency under adverse conditions.

5.3. Results of flying control with evaluation metrics

Figure 12 presents the trajectories, with the first row depicting three flying control "left," "down," and "back" trajectories under normal conditions, while the second row illustrates the same flying control under the effects of sensor noise for sigma values of 0.1, 0.3, and 0.5. The ability of the drone

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

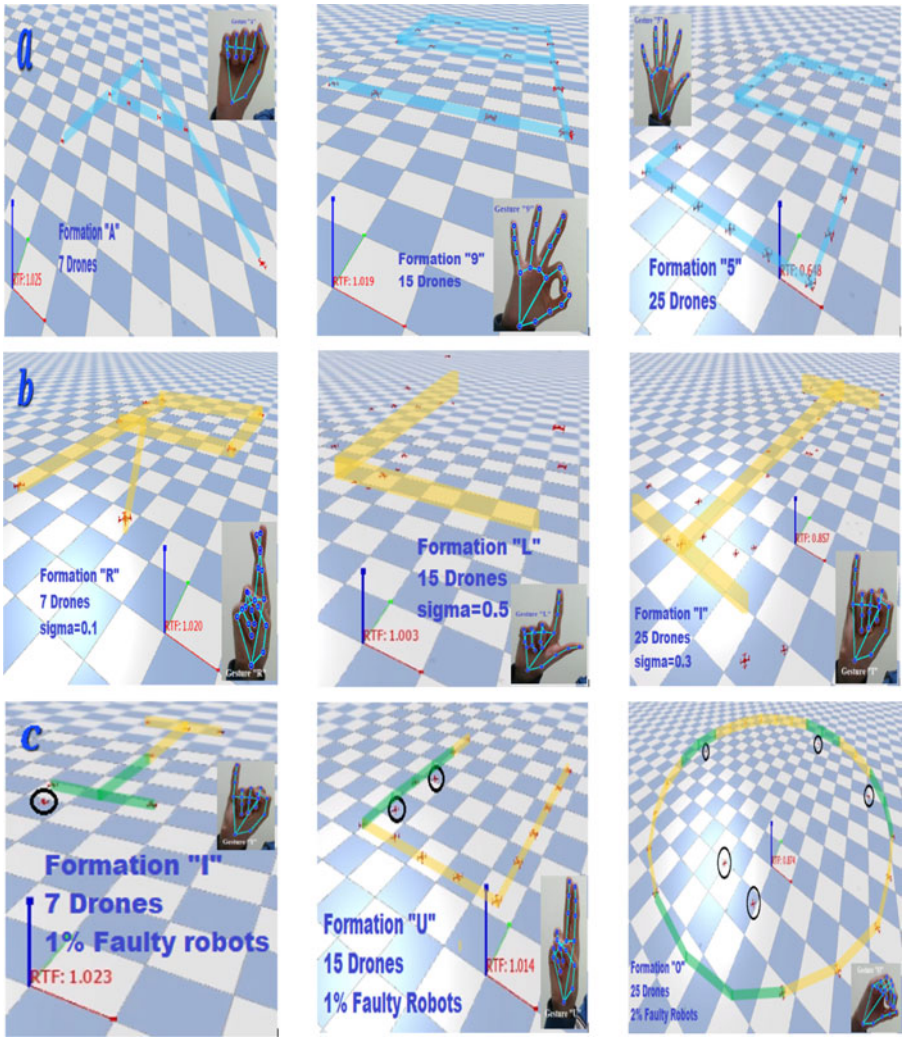


Figure 9. *Simulation results of swarms composed of 7, 15, and 25 drones performing gesture-based formations under three conditions: normal operation (a), sensor noise (b), and faulty robots (c).*

swarm to maintain formation is highly dependent on noise conditions. Low noise allows for near-perfect alignment, while higher noise leads to significant misshaping of the swarm configuration.

5.3.1. Results with normal conditions

Based on the data presented in Table VIII, which includes the Normal column, and the average accomplishment time for each formation illustrated in Figure 10(a) (first row), we can analyse the results. Additionally, Table IX (column Normal) and Figure 11(a) (first column) provide insights into the unified mean error calculation of positions. Furthermore, Table X (column Normal) outlines the standard deviation and coefficient of variation values. This comprehensive analysis leads us to the following conclusions:

For 7 drones, the control commands take a completion time ranging from 15.78 to 16.41 s. Command "down" exhibits the most consistent completion time. The maximal value of SD and CV does not exceed 0.04 for SD and 0.0024 for CV, revealing that the commands are achieved in a stable time. The UMCE values demonstrate high path optimisations with a maximal error of 1E-06, with a collision rating almost null for all commands. However, the increasing swarm size leads to an increase in time completion for

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table VIII. *Average control accomplishment time in seconds for different swarm size 7-15-25 in three cases: normal conditions, effect of sensor noise (with σ 0.1, 0.3 and 0.5) and effect of 1% and 2% of faulty robots.*

		Noise			Faulty robots	
Command	**Normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**
			7 drones			
"up"	15.8	15.82	16.38	16.40	25.55	25.56
"down"	15.78	15.81	16.37	16.39	25.54	25.57
"left"	15.79	15.81	16.40	16.37	25.57	25.54
"right"	15.80	15.82	15.87	16.38	25.56	25.55
"go"	15.82	15.81	16.40	16.37	25.57	25.58
"back"	15.82	15.81	16.40	16.37	25.56	25.56
"takeoff"	15.80	15.83	16.41	16.39	25.58	25.6
"stop"	16.41	16.38	16.43	16.42	26.15	26.21
			15 drones			
"up"	16.01	15.96	16.49	16.55	25.83	25.76
"down"	16.04	15.91	16.49	16.57	25.80	25.77
"left"	16.00	15.93	16.50	16.56	25.84	25.77
"right"	16.01	15.91	16.49	16.55	25.79	25.83
"go"	16.01	15.90	16.49	16.56	25.79	25.81
"back"	16.03	15.90	16.49	16.55	25.77	25.78
"takeoff"	16.04	15.91	16.49	16.55	25.79	25.77
"stop"	16.62	16.56	16.55	16.55	26.48	26.42
			25 drones			
"up"	16.10	16.09	16.85	16.68	26.31	26.03
"down"	16.08	16.13	16.84	16.7	26.53	26.15
"left"	16.1	16.08	16.85	16.77	26.26	26.19
"right"	16.07	16.08	16.85	16.71	26.21	26.21
"go"	16.1	16.08	16.66	16.72	26.43	26.21
"back"	16.07	16.08	16.71	16.71	26.08	26.13
"takeoff"	16.08	16.07	16.7	16.67	26.65	28.53
"stop"	17.85	18.14	18.09	18.53	28.34	28.46

15 and 25 drones. Ranging between 16 and 17.85 s. The command "left" has the best accuracy. The command "stop" is the least efficient due to the landing forces. A slight increase of SD and CV across commands for both 15 and 25 drones, varying from 0.13 to 0.14 for SD and 0.76% and 0.84% for CV, indicating a very low variation in time completion. UMCE value does not exceed 1E-6, which means a highly accurate system ensures safety by avoiding collision and maintaining task completion efficiently.

5.3.2. Effects of sensor noises

Analysing the information from Table VIII, specifically the "Noise" column, along with Figure 10(a) (second row), which illustrates the average accomplishment time for each formation, provides valuable insights. Additionally, Table IX (column "Noise") and Figure 11(a) (second column) present the unified mean error calculations for positions. Table X (column "Noise") offers details on the SD and CV. Together, these elements enable us to draw the following conclusions: For control commands under noise levels, it took more time to achieve the task. Varying from 15.81 to 18.53 s for the three levels of noise. Commands show a consistent completion time with slight variability, demonstrated by SD and CV. However, these are still considered less accurate than normal conditions. The UMCE values

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

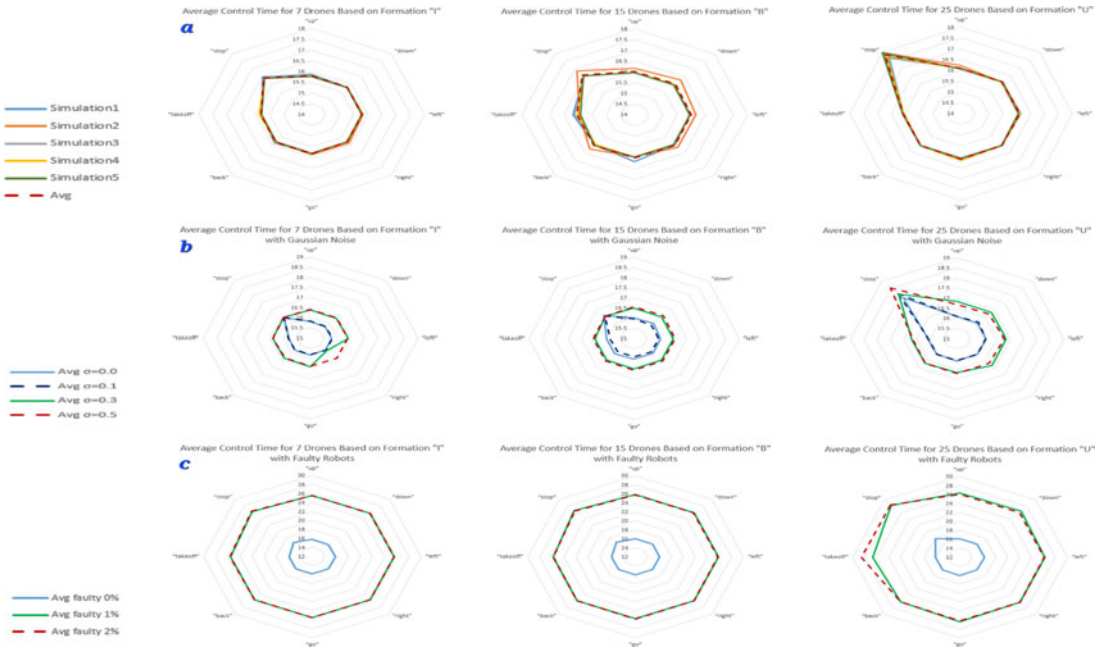


Figure 10. *Average control accomplishment time (seconds) for swarms composed of 7, 15, and 25 drones under different formation tasks. The results compare performance under three scenarios: normal conditions, sensor noise levels ($\sigma = 0.1, 0.3, 0.5$), and faulty robot ratios (1% and 2%), illustrating the effect of uncertainty and faults on control efficiency.*

demonstrate high values compared with normal conditions, which suggests that the increase in noise levels increases the complexity of the path, with a range of error between 0.11 and 0.79, with a collision rating almost null for all commands. However, the increasing swarm size increases the time to completion for 15 and 25 drones. Ranging between 16 and 17.85 s. The command "stop" is the least efficient due to the landing forces.

5.3.3. Effects of faulty robots

By examining the data from Table VIII, particularly the "Faulty Robots" column, and referring to Figure 10(a) (third row), which shows the average accomplishment time for each formation, we gain important insights. In addition, Table IX (column "Faulty Robot") and Figure 11(a) (third column) provide the unified mean error calculations for positions. Table X (column "Faulty Robots") contains information on the standard deviation and coefficient of variation. Together, these components allow us to reach the following conclusions:

The faulty robots scenario takes the longest completion time. The command "up" achieved 25.55 s for 7 drones with one faulty robot, while the command "stop" recorded the least efficient time of 28.46 s across different swarm size scenarios. The UMCE values are consistent with those in the normal cases, indicating the paths are well-optimised. The cumulative position error does not exceed 1E-6, demonstrating high collision-avoidance application and safety ensuring. Low variability according to the CV values suggests consistent completion time. However, it is noted that with the increasing swarm size, both time completion and error in achieving final positions increase.

Cumulatively, these results point to the necessity of continued monitoring and maintenance of drone fleets in order to achieve optimum performance under realistic environments, especially dynamic ones where communication and coordination are required for operational success.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table IX. *Unified Mean Cumulative Error (UMCE) calculating for different swarm size 7-15-25 in three cases: normal conditions, effect of sensor noise (with σ 0.1, 0.3 and 0.5) and effect of 1% and 2% of faulty robots.*

		Noise			Faulty robots	
Command	**Normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**
			7 drones			
"up"	1.000000000140E-06	1.581776268801E-01	4.745334286673E-01	7.908891615694E-01	1.000000000140E-06	1.000000000140E-06
"down"	1.000000000033E-06	1.581777659409E-01	4.745334400977E-01	7.908891729998E-01	1.000000000033E-06	1.000000000033E-06
"left"	1.000000000079E-06	1.581775114526E-01	4.745331392250E-01	7.908889056088E-01	1.000000000079E-06	1.000000000079E-06
"right"	1.000000000045E-06	1.581781947338E-01	4.745336091547E-01	7.908892850119E-01	1.000000000045E-06	1.000000000045E-06
"go"	9.999999999812E-07	1.581774451635E-01	4.745330321441E-01	7.908889100483E-01	9.999999999812E-07	9.999999999812E-07
"back"	9.999999999902E-07	1.581781418499E-01	4.745333607824E-01	7.908891529721E-01	9.999999999902E-07	9.999999999902E-07
"takeoff"	9.999999999177E-07	1.581776268801E-01	4.745334286673E-01	7.908891615694E-01	9.999999999177E-07	9.999999999177E-07
"stop"	9.999999999938E-07	1.327404985529E-01	3.982219374403E-01	6.637033920396E-01	9.999999999938E-07	9.999999999938E-07
			15 drones			
"up"	1.000000000140E-06	1.532943463296E-01	4.598828064942E-01	7.664712824519E-01	1.000000000140E-06	1.000000000140E-06
"down"	1.000000000033E-06	1.532940763148E-01	4.598825274668E-01	7.664710034246E-01	1.000000000033E-06	1.000000000033E-06
"left"	1.000000000009E-06	1.532940435138E-01	4.598825042061E-01	7.664712149113E-01	1.000000000009E-06	1.000000000009E-06
"right"	1.000000000081E-06	1.532944157977E-01	4.598826263320E-01	7.664713289188E-01	1.000000000081E-06	1.000000000081E-06
"go"	1.000000000021E-06	1.532941036052E-01	4.598825086941E-01	7.664712131801E-01	1.000000000021E-06	1.000000000021E-06
"back"	9.999999999817E-07	1.532943576765E-01	4.598826139541E-01	7.664712784398E-01	9.999999999817E-07	9.999999999817E-07
"takeoff"	9.999999999177E-07	1.532943463296E-01	4.598828064942E-01	7.664712824519E-01	9.999999999177E-07	9.999999999177E-07
"stop"	9.999999999938E-07	1.142437150482E-01	3.427312495067E-01	5.712188150817E-01	9.999999999938E-07	9.999999999938E-07
			25 drones			
"up"	1.000000000140E-06	1.555899736391E-01	4.667700811801E-01	7.779502073194E-01	1.000000000140E-06	1.000000000140E-06
"down"	1.000000000033E-06	1.555899549390E-01	4.667699986023E-01	7.779501729802E-01	1.000000000033E-06	1.000000000033E-06
"left"	1.000000000062E-06	1.555898617918E-01	4.667698556440E-01	7.779501392279E-01	1.000000000062E-06	1.000000000062E-06
"right"	1.000000000000E-06	1.555900777233E-01	4.667699086726E-01	7.779502138239E-01	1.000000000000E-06	1.000000000000E-06
"go"	1.000000000044E-06	1.555899389698E-01	4.667699554456E-01	7.779501729802E-01	1.000000000044E-06	1.000000000044E-06
"back"	1.000000000044E-06	1.555900060546E-01	4.667699478674E-01	7.779502077360E-01	1.000000000044E-06	1.000000000044E-06
"takeoff"	9.999999999177E-07	1.555899736391E-01	4.667700811801E-01	7.779502073194E-01	9.999999999177E-07	9.999999999177E-07
"stop"	9.999999999938E-07	1.107168550599E-01	3.321508060682E-01	5.535846256499E-01	9.999999999938E-07	9.999999999938E-07

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Table X. *Standard deviation (SD) and coefficient of variation (CV) calculating for different swarm size 7-15-25 in three cases: normal conditions, effect of sensor noise (with σ 0.1, 0.3 and 0.5) and effect of 1% and 2% of faulty robots.*

		SD noise			SD faulty robots			CV noise			CV faulty robots	
Command	**SD normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**	**CV normal**	$\sigma = 0.1$	$\sigma = 0.3$	$\sigma = 0.5$	**1%**	**2%**
7 drones												
"up"	0.03	0.02	0.03	0.01	0.00	0.02	0.0020	0.0014	0.0017	0.0007	0.0002	0.0007
"down"	0.00	0.00	0.00	0.02	0.01	0.04	0.0003	0.0002	0.0002	0.0012	0.0003	0.0017
"left"	0.02	0.00	0.00	0.01	0.04	0.01	0.0014	0.0002	0.0002	0.0006	0.0016	0.0002
"right"	0.04	0.01	0.01	0.01	0.04	0.01	0.0025	0.0007	0.0004	0.0008	0.0014	0.0003
"go"	0.03	0.00	0.01	0.01	0.04	0.05	0.0016	0.0001	0.0008	0.0004	0.0014	0.0018
"back"	0.04	0.01	0.01	0.01	0.03	0.03	0.0023	0.0007	0.0005	0.0004	0.0013	0.0012
"takeoff"	0.04	0.06	0.01	0.02	0.04	0.00	0.0023	0.0036	0.0008	0.0011	0.0015	0.0002
"stop"	0.03	0.00	0.01	0.01	0.01	0.01	0.0021	0.0002	0.0008	0.0006	0.0005	0.0005
15 drones												
"up"	0.08	0.01	0.00	0.01	0.05	0.08	0.0049	0.0005	0.0001	0.0003	0.0018	0.0029
"down"	0.13	0.02	0.01	0.00	0.05	0.08	0.0084	0.0012	0.0004	0.0003	0.0017	0.0031
"left"	0.08	0.04	0.01	0.01	0.01	0.06	0.0052	0.0023	0.0009	0.0008	0.0005	0.0024
"right"	0.07	0.02	0.01	0.00	0.07	0.11	0.0046	0.0014	0.0004	0.0003	0.0027	0.0044
"go"	0.08	0.01	0.01	0.01	0.07	0.09	0.0052	0.0004	0.0004	0.0007	0.0028	0.0034
"back"	0.11	0.00	0.01	0.00	0.08	0.04	0.0069	0.0003	0.0004	0.0003	0.0030	0.0014
"takeoff"	0.10	0.02	0.01	0.00	0.09	0.07	0.0064	0.0013	0.0007	0.0003	0.0035	0.0029
"stop"	0.13	0.02	0.01	0.01	0.10	0.03	0.0079	0.0012	0.0007	0.0005	0.0039	0.0011
25 drones												
"up"	0.06	0.04	0.05	0.01	0.22	0.06	0.0035	0.0026	0.0028	0.0007	0.0085	0.0021
"down"	0.01	0.07	0.02	0.01	0.21	0.10	0.0008	0.0041	0.0012	0.0004	0.0079	0.0040
"left"	0.03	0.01	0.08	0.13	0.13	0.09	0.0021	0.0005	0.0045	0.0077	0.0051	0.0033
"right"	0.00	0.02	0.05	0.06	0.16	0.13	0.0002	0.0013	0.0027	0.0035	0.0063	0.0049
"go"	0.04	0.03	0.01	0.09	0.32	0.05	0.0024	0.0017	0.0003	0.0053	0.0121	0.0018
"back"	0.00	0.02	0.08	0.05	0.10	0.18	0.0003	0.0014	0.0045	0.0030	0.0039	0.0069
"takeoff"	0.02	0.03	0.04	0.00	0.29	0.27	0.0015	0.0021	0.0024	0.0003	0.0108	0.0095
"stop"	0.14	0.05	0.19	0.32	0.53	0.46	0.0076	0.0025	0.0108	0.0170	0.0188	0.0162

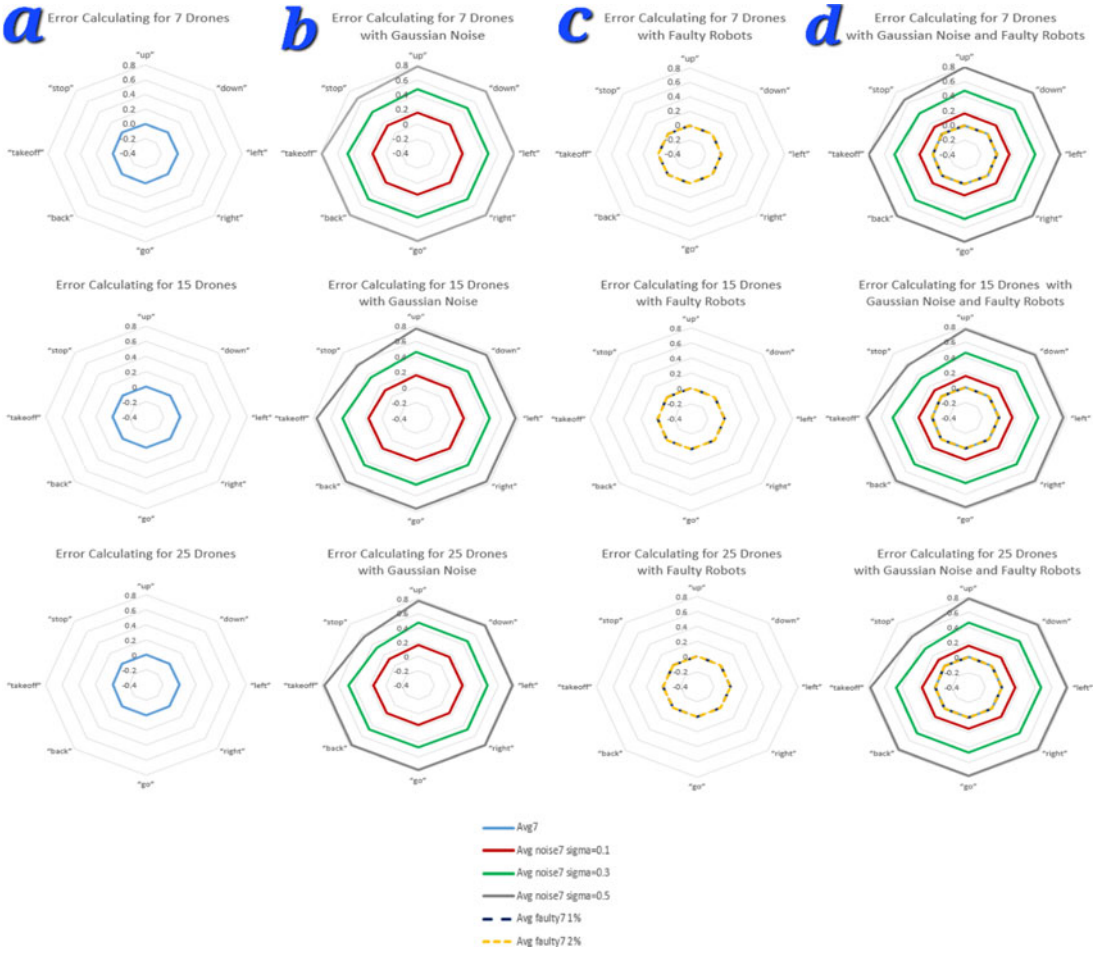


Figure 11. *Averaged error for swarms of 7, 15, and 25 drones executing gesture-based control commands under three scenarios: normal operation, sensor noise with Gaussian variance ($\sigma = 0.1, 0.3, 0.5$), and faulty robot ratios of 1–2%.*

The overall accomplishment time is consistent in both normal conditions and with noise scenarios, increasing with the increase of drone size, which reveals a scalability relative. The introduction of faulty robots significantly increases the accomplishment times, nearly doubling compared to normal conditions. This indicates a substantial sensitivity to faults. SD and CV values under the three conditions are relatively low but can vary slightly. This shows consistency and reliability in formation accomplishment, suggesting that while noises and faults introduce some variability, the collision avoidance mechanism can still maintain a degree of precision. Predicted collision rates under the three conditions are generally low, not exceeding a 7% rate. The overall UMCE values suggest that formation efficiency and safety are guaranteed by collision avoidance. Some formations, like "2" (normal scenario for 7 drones), "R" ($\sigma =$ 0.1 for 7 drones), "C" ($\sigma = 0.5$ for 7 drones), "U" ($\sigma =$ 0.1 for 15 drones), "L" and "1" ($\sigma = 0.5$ for 15 drones) show higher UMEC, indicating that noise levels can impact the effectiveness of path due to the applied forces that adjust positions to prevent collisions.

Under ideal conditions, the swarm operates with high stability and efficiency, with minimal variability observed in movement and completion times, and collision rates remain close to zero. Introduction of noise leads to slightly increased completion times, more complex and less direct flight paths. Overall swarm performance remains largely unaffected. Faulty scenarios result in over 50% increase in task completion times and elevated risk of collisions, especially as swarm size grows. Faulty drones are

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

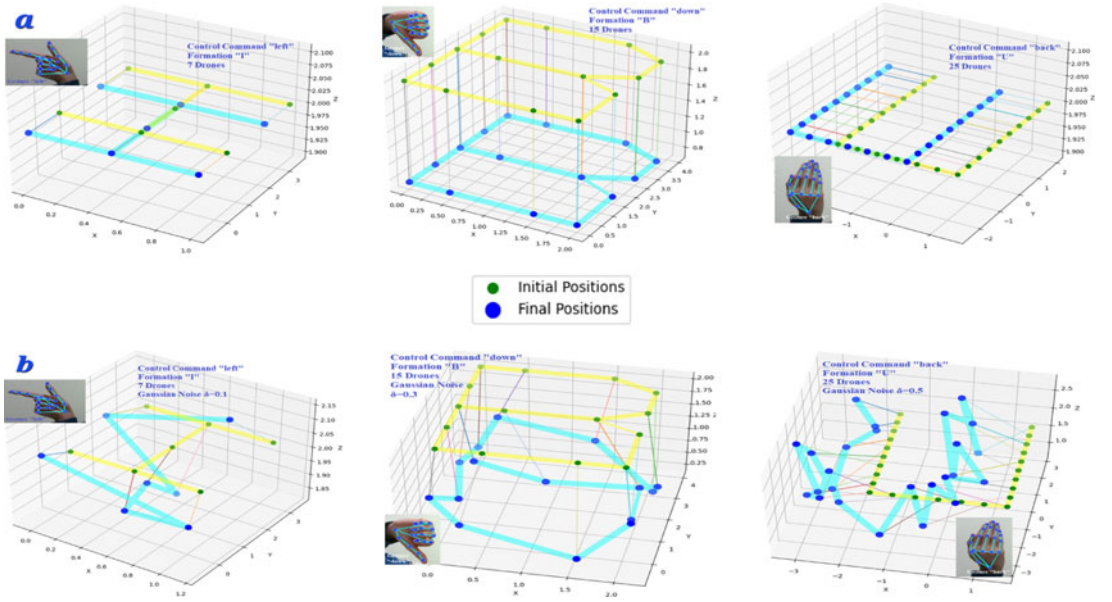


Figure 12. *Trajectories of swarms composed of 7, 15, and 25 drones executing gesture-based flight commands under different sensor noise levels ($\sigma = 0.1, 0.3, 0.5$). The figure illustrates the impact of noise on the evolution of swarm motion during formation execution.*

intentionally paused for 5 s with no trajectory updates. During these pauses, the rest of the swarm must wait, resulting in delays and disrupted flight orientation.

5.4. Statistical analysis of the impact of swarm scale and scenario on controller performance

In this section, an overall statistical analysis integrating data findings across the formation and flying study types is conducted to assess the overarching effects of swarm scale and scenario on the performance metrics: Accomplished Time, UMCE, SD, and CV. A two-way Analysis of Variance (ANOVA) was executed for each performance metric, treating the studied swarm scale (7, 15, 25 drones) and scenarios (Normal, Noise $\sigma = 0.1$, Noise $\sigma = 0.3$, Noise $\sigma = 0.5$, % faulty robots = 1, and % faulty robots = 2) as independent factors. Metrics demonstrating statistically significant interaction effects ($p < 0.05$) subsequently underwent Tukey's Honestly Significant Difference (HSD) post-hoc tests to delineate specific pairwise mean differences among combined swarm scale and scenario groups.

The overall two-way ANOVA elucidated differential influences of swarm scale and scenario on controller performance. For Accomplished Time, both swarm scale ($F(2, 450) = 125.73, p < 0.001$) and scenario ($F(5, 450) = 1854.79, p < 0.001$) exhibited highly significant main effects, alongside a significant interaction effect ($F(10, 450) = 2.13, p = 0.0209$), indicating that the effect of swarm scale on task accomplishment time is contingent upon the scenario conditions, a crucial consideration for mission planning. In contrast, UMCE demonstrated a highly significant main effect for scenario ($F(5, 450) = 169.10, p < 0.001$), but neither swarm scale ($F(2, 450) = 0.15, p = 0.858$) nor its interaction with scenario ($F(10, 450) = 1.30, p = 0.225$) reached statistical significance. This implies that while scenarios significantly alter UMCE, this impact is largely consistent across varying swarm scales, suggesting an additive rather than interactive contribution of swarm size to UMCE metrics under these controller types. For both SD and CV, swarm scale (SD: $F(2, 450) = 165.72, p < 0.001$; CV: $F(2, 450) = 147.05, p < 0.001$) and scenario (SD: $F(5, 450) = 4.97, p < 0.001$;$CV$:$F(5, 450) = 5.45, p < 0.001$) demonstrated highly significant main effects. Furthermore, highly significant interaction effects were identified (SD: $F(10, 450) = 6.45, p = 2.54e - 09$; CV: $F(10, 450) = 6.16, p = 7.79e - 09$), indicating that the

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

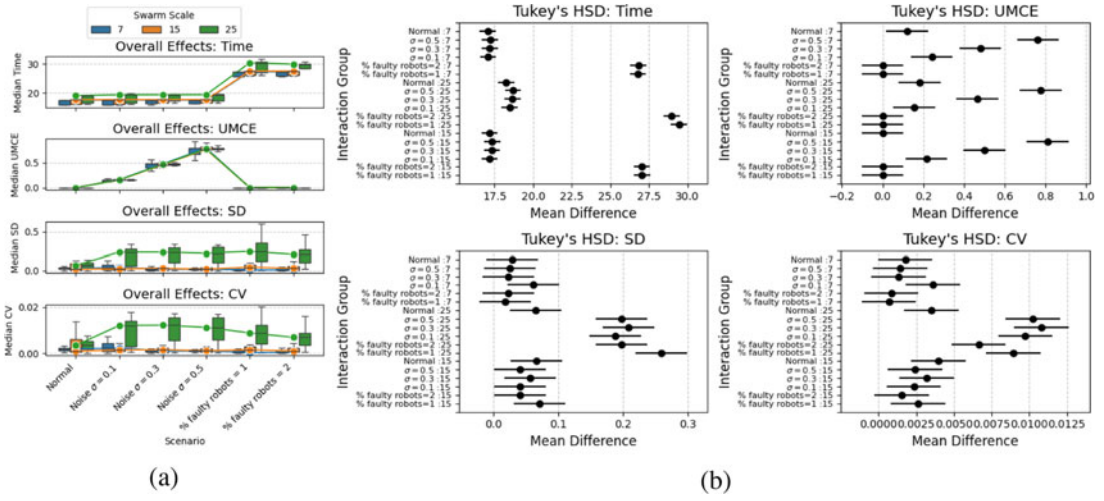


Figure 13. *Overall interaction effects of swarm size and scenario on performance metrics. (a) Analysis of interaction effects between swarm scale and scenario conditions. (b) Post hoc Tukey's HSD pairwise comparisons for performance metrics.*

variability of controller performance is non-trivially influenced by the specific combination of swarm size and operational conditions.

Furthermore, post hoc Tukey's HSD tests, conducted for Accomplished Time, SD, and CV due to their significant interaction effects, revealed extensive significant pairwise differences across combined swarm scale and scenario groups. These comparisons, visually represented in Figure 13(a), primarily highlighted substantial deviations when contrasting % faulty robots scenarios against Normal or Noise σ conditions. For instance, faulty robot scenarios consistently led to significantly higher mean Accomplished Time and increased performance variability (SD, CV), with swarm scale critically modulating the magnitude of these effects. These findings underscore the context-dependent robustness of swarm controllers. Conversely, for UMCE, where the global interaction was not significant, Tukey's HSD still identified significant differences between Normal/Noise σ and % faulty robots groups. However, comparisons between different swarm scale levels within similar scenarios were less frequently significant (Figure 13(b)), corroborating the non-significant interaction effect and suggesting that controller design for UMCE may leverage independent optimisation of scenario-specific parameters rather than complex scale-interaction adjustments.

5.5. Limitations

The proposed framework is limited to simulation-based validation only; using CrazyFlyt, no real experiments have been conducted with physical drones. Real-world effects such as actuator dynamics, communication delays, and environmental disturbances are not fully captured. In addition, the system has been evaluated on a limited set of gesture classes and swarm sizes. Scaling to large swarms and more complex commands may introduce additional challenges. User-dependent variability in gesture execution and sensitivity to environmental conditions such as lighting variations and occlusions have not been explicitly addressed. While fault tolerance is incorporated, robustness under higher fault rates and more dynamic real-world environments remains to be further investigated.

6. Conclusion

This study focuses on developing an intuitive gesture-based recognition system for controlling swarm robotics, specifically for formation and navigation tasks. By mapping human gestures directly to swarm

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

commands, the system aims to streamline HSI and enhance operational efficiency. Multiple neural network architectures were evaluated, including LSTM, BiLSTM, GRU, BiGRU, ResNet101, and MobileNetV2. BiGRU achieved a notable accuracy of 99.4% in gesture recognition. Recognised gestures are directly linked to corresponding swarm formations and commands. The experimental validation is testing under three conditions: normal scenarios, Gaussian noise at varying levels (0.1, 0.3, 0.5), and faulty robot cases (1% and 2% malfunction rate). All experiments were conducted using the CrazyFlyt simulator. The key outcomes of these experiments are high reliability under normal conditions, and performance with larger swarms (7, 15, 25 drones) shows only a slight decrease in accuracy. For the low noise levels have minimal impact; higher noise introduces noticeable degradation. Faulty robots delay the formation and navigation achievement, causing longer completion times and raising collision risks, especially as the swarm size increases. Nevertheless, the integration of collision-avoidance mechanisms maintains safety and operational consistency, even when faults are present.

Despite these promising results, the study has certain limitations. Most importantly, the evaluation is limited to simulation (CrazyFlyt), and no physical drone experiments have been conducted. This means that real-world effects such as actuator dynamics, communication delays, and environmental disturbances are not fully captured. Second, the current system handles a limited set of gesture classes and swarm sizes; scaling to larger swarms or more complex commands may introduce additional challenges. In addition, potential variability in gesture recognition across different users, as well as the sensitivity of the vision-based interface to environmental conditions such as lighting variations and occlusions, have not been explicitly addressed. Third, while fault tolerance is incorporated, performance under higher fault rates or more dynamic environments remains to be further investigated.

For future extensions, the primary direction is real-world validation using physical drone platforms, even at a small scale, to evaluate practical performance and robustness. Future work will also extend evaluation beyond accuracy to include latency, computational cost, and energy efficiency, with comparisons to state-of-the-art methods. In addition, more gesture classes will be incorporated to broaden the range of swarm commands and enable more complex manoeuvres. Further improvements in fault-tolerant control strategies will also be essential to handle severe robot failures and maintain stable swarm coordination. Collectively, these enhancements are expected to improve the flexibility and intuitiveness of the control system, foster more effective human–swarm interaction, and increase operational efficiency across diverse real-world applications.

To ensure reproducibility and transparency, the complete implementation, trained models, and simulation framework are publicly available in a dedicated GitHub repository.[2]

Acknowledgements. The authors would like to acknowledge the LABRI-SBA Research Laboratory for its support and for providing the necessary resources to conduct this research.

Author contributions. **Fatna Bent Ennebi Benabbou:** Conceptualization, Methodology, Software, Development of Gesture Recognition and Swarm Formation Frameworks, Implementation, Simulation Design and Execution, Formal Analysis, Investigation, Data Curation, Visualization, Writing – Original Draft Preparation. **Belkacem Khaldi:** Methodology, Validation, Formal Analysis, Supervision, Writing – Review & Editing. All authors reviewed, edited, and approved the final manuscript. **Sidi Mohammed Benslimane:** Conceptualization, Methodology, Supervision, Resources, Funding Acquisition, Project Administration, Writing – Review & Editing.

Financial support. This work was supported by the "Multimodal Communication and Time Well Spent with Smartphones in the Context of the Internet of Things" project Grant Agreement C00L07ES220120220002 under the University Training Research Program (PRFU).

Competing interests. The authors declare no conflicts of interest exist.

Ethical standards. Not applicable.

[2] https://github.com/BENABBOU666/swarm-gesture-control-bigru-crazyflyt.

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

Use of Artificial Intelligence (AI). The authors used OpenAI ChatGPT (GPT-5.5, OpenAI, https://chatgpt.com) during the preparation of this manuscript solely for language editing, grammar correction, and improvement of readability. The tool was used between 2025 and 2026.

All scientific content, including study design, methodology, experiments, results, interpretation, and conclusions, was developed, validated, and verified exclusively by the authors. No artificial intelligence tools were used for data collection, data analysis, simulation generation, or decision-making processes.

The authors confirm that the use of AI did not introduce bias or affect the scientific integrity of the work. The authors assume full responsibility for the content of the manuscript in accordance with the publication standards of Cambridge University Press journals, including *Robotica*.

References

[1] M. Brambilla, E. Ferrante, M. Birattari and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intell.* **7**(1), 1–41 (2013).

[2] B. Khaldi and F. Cherif, "An overview of swarm robotics: Swarm intelligence applied to multi-robotics," *Int. J. Comput. Appl.* **126**(2), 31–37 (2015).

[3] A. Kolling, S. Nunnally and M. Lewis, "Towards Human Control of Robot Swarms," *Proceedings of the Seventh Annual ACM/IEEE International Conference On Human–Robot Interaction 2012, HRI'12* (Association for Computing Machinery, New York, NY, USA, 2012) pp. 89–96.

[4] A. Kolling, P. Walker, N. Chakraborty, K. Sycara and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE T Hum.-MACH Syst.* **46**(1), 9–26 (2016).

[5] S. Nunnally, P. Walker, M. Lewis, N. Chakraborty and K. Sycara, "Using Haptic Feedback in Human Robotic Swarms Interaction," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. **57** (2013) pp. 1047–1051.

[6] S. O. Sachidanandam, S. Honarvar and Y. Diaz-Mercado, "Effectiveness of Augmented Reality for Human Swarm Interactions," *2022 International Conference on Robotics and Automation (ICRA)* (IEEE, 2022) pp. 11258–11264.

[7] J. Nagi, A. Giusti, L. M. Gambardella and G. A. Di Caro, "Human-Swarm Interaction Using Spatial Gestures," *2014 IEEE/RSJ International Conference On Intelligent Robots and Systems* (IEEE, 2014) pp. 3834–3841.

[8] C. Creed, M. Al-Kalbani, A. Theil, S. Sarcar and I. Williams, "Inclusive augmented and virtual reality: A research agenda," *Int. J. Hum.–Comput. Interact.* **40**(20), 6200–6219 (2024).

[9] F. B. E. Benabbou, B. Khaldi and S. M. Benslimane, "Human-Swarm Interactions: A Bibliometric Analysis Based on Citespace," *First International Conference on Big Data, IOT, Web Intelligence and Applications (BIWA)* (IEEE, 2022) pp. 53–58.

[10] A.-I. Siean, B.-C. Gradinaru, O.-I. Gherman, M. Danubianu and L.-D. Milici, "Opportunities and challenges in human-swarm interaction: Systematic review and research implications," *Int. J. Adv. Comput. Sci. Appl.* **14**(4), 896–902 (2023).

[11] P. N. Huu, P. D. L. Hong, D. D. Dang, B. V. Quoc, C. N. L. Bao and Q. T. Minh, "Proposing Hand Gesture Recognition System Using MediaPipe Holistic and LSTM," *International Conference On Advanced Technologies for Communications (ATC)* (IEEE, 2023).

[12] Z. Kakish, S. Vedartham and S. Berman, Towards decentralized human-swarm interaction by means of sequential hand gesture recognition. arXiv preprint arXiv:2102.02439 (2021).

[13] E. Kučera, O. Haffner, M. Shevska and D. Janecký, "Evaluation of Motion-Capture Suit Data and Gesture Recognition Using LSTM and GRU Neural Networks," *2025 Cybernetics & Informatics (K&I)* (IEEE, 2025) pp. 1–6.

[14] R. Wang and C. Sun, "A Gesture Recognition and Drone Control System Based on Residual Neural Network," *2023 IEEE International Conference on Mechatronics and Automation (ICMA)* (2023) pp. 25–30.

[15] F. Gongor and O. Tutsoy, "Humanoid Robots Learn and Recognize Seven Facial Emotions with ANN," *International Science and Academic Congress, INSAC18* (2018).

[16] R. Raj and I. Demirkol, "An improved facial emotion recognition system using convolutional neural network for the optimization of human robot interaction," *Sci. Rep-UK* **15**(1), 38940 (2025).

[17] O. Tutsoy, *A Review of Recent Advancements in Deep Machine Learning, Artificial Intelligence, Object Detection, and Human-Robot Interactions Approaches For Assistive Robotics Ph.D., Fatma Gongor., 27* (2023) pp. 45–56.

[18] C. Kawatsu, F. Koss, A. Gillies, A. Zhao, J. Crossman, B. Purman, D. Stone and D. Dahn, "Gesture Recognition for Robotic Control Using Deep Learning," *SAE Technical Paper Series 2024, volume 1, 2101 Wilson Blvd, Suite 700, Arlington, VA 22201, USA* (National Defense Industrial Association, 2017).

[19] Y.-P. Yeh, S.-J. Cheng and C.-H. Shen, "Research on Intuitive Gesture Recognition Control and Navigation System of UAV," *2022 IEEE 5th International Conference on Knowledge Innovation and Invention (ICKII)* (IEEE, 2022).

[20] A. Naderolasli, "Indirect self-tuning controller for a two degree of freedom tracker model," *Int. J. Veh. Auton. Syst.* **16**(1), 15 (2021).

[21] A. Naderolasli, "Constrained super-twisting sliding-mode controller for beam and ball nonlinear system stabilization with limited field-of-view constraints," *Iran. J. Sci. Technol. Trans. Electr. Eng.* **49a**(4), 1645–1659 (2025a).

[22] A. Naderolasli, "Stabilization of plate and ball nonlinear system with limited field-of-view sensors using constrained adaptive sliding-mode controller," *Adv. Control Appl.: Eng. Ind. Syst.* **7b**(3), e70025 (2025b).

[23] H. Shirzadfar, "Different applications of the gyroscope sensors data fusion in distinctive systems: An extended Kalman filter approach," *Computacion y Sistemas* **28**(1), 189–196 (2024).

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press

[24] T. Almuzaini and A. Savkin, "AUV trajectory planning for optimized sensor data collection in internet of underwater things," *Future Internet* **17**(7), 293 (2025).
[25] M. A. Hsieh, V. Kumar and L. Chaimowicz, "Decentralized controllers for shape generation with robotic swarms," *Robotica* **26**(5), 691–701 (2008).
[26] L. F. Sanchez, H. Abaunza and P. Castillo, "User–robot interaction for safe navigation of a quadrotor," *Robotica* **38**(12), 2189–2203 (2020).
[27] H. Huang, D. Wu, Z. Liang, F. Sun and M. Dong, "Virtual interaction and manipulation control of a hexacopter through hand gesture recognition from a data glove," *Robotica* **40**(12), 4375–4387 (2022).
[28] Y. Li, Y. Wang, Y. Mo, Q. Zhu, Z. Miao and Y. Luo, "Natural Interactive Control Method for Unmanned Swarm Systems Based on Gesture Recognition," *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (IEEE, 2022).
[29] S. Khaksar, L. Checker, B. Borazjan and I. Murray, "Design and evaluation of an alternative control for a quad-rotor drone using hand-gesture recognition," *Sensors (Basel)* **23**(12), (2023).
[30] G. Khen, D. Zhao and J. Baca, "Intuitive Human-Swarm Interaction with Gesture Recognition and Machine Learning," *The Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc'23)* (ACM, 2023) pp. 1–4.
[31] M. H. Zafar, E. F. Langås and F. Sanfilippo, "Real-Time Gesture-based Control of a Quadruped Robot Using a Stacked Convolutional Bi-Long Short-Term Memory (Bi-LSTM) Neural Network," *2024 10th International Conference on Automation, Robotics and Applications (ICARA)* (IEEE, 2024).
[32] A. I. Google, Mediapipe gesture recognizer (2025). [Accessed: 19 August 2025].
[33] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Rob. Auton. Syst.* **27**(3), 171–194 (1999).
[34] V. Gazi and K. Passino, "A Class of Attraction/Repulsion Functions for Stable Swarm Aggregations," *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. **3** (2002) pp. 2842–2847.
[35] J. J. Tai, J. Wong, M. Innocente, N. Horri, J. Brusey and S. K. Phang, "PyFlyt–UAV simulation environments for reinforcement learning research (2023)". arXiv preprint arXiv:2304.01305.
[36] Z. Jalilibal, A. Amiri, P. Castagliola and M. B. C. Khoo, "Monitoring the coefficient of variation: A literature review," *Comput. Ind. Eng.* **161**, Article no. 107600 (2021).
[37] K. S. Parmar and R. Bhardwaj, "Water quality management using statistical analysis and time-series prediction model," *Appl. Water Sci.* **4**(4), 425–434 (2014).
[38] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Automat. Control* **51**(3), 401–420 (2006).

Cite this article: F. B. E. Benabbou, B. Khaldi and S. M. Benslimane (2026). "Human-swarm interaction for formation/flying control by enhanced vision-based hand gesture recognition system", Robotica **44**, 1573–1607. https://doi.org/10.1017/S0263574726103725

https://doi.org/10.1017/S0263574726103725 Published online by Cambridge University Press