# Collisionless Fast Pattern Formation Mechanism for Dynamic Number of UAVs

Gunasekaran Raja, *Senior Member, IEEE*, Saran V S, Sudha Anbalagan, Ali Kashif Bashir, *Senior Member, IEEE*, Muhammad Imran, *Senior Member, IEEE*, Nidal Nasser, *Senior Member, IEEE*

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322085

***Abstract*—Unmanned Aerial Vehicle (UAV) is an emerging technology that assists in various automated activities where human involvement is minimal. Though individual UAVs are extremely useful entities, their productivity can further be increased by deploying multi-UAVs. Pattern formation among multi-UAVs is one of the key functionalities in a swarm environment that is essential for several UAV missions namely military expedition, search and rescue operations, drone based delivery mechanisms etc. In this paper, to facilitate pattern formation among UAVs in an effective manner, a Time-Interleaved Pattern Formation (TIPF) Mechanism is proposed. The existing systems work for a fixed number of drones whose pattern switching mechanisms are preprogrammed. However, the TIPF mechanism enables switching patterns among dynamic number of drones (UAVs) on the fly by inducing a small delay between each UAV movement. The TIPF mechanism avoids collision, which occurs due to the simultaneous movement of UAVs. The proposed TIPF mechanism encompasses a Centralised Coordinate Calculation (CCC) algorithm to easily calculate the coordinates of UAVs in a given pattern. Further, this mechanism has also been simulated and tested in our proposed virtual IP based Software In The Loop (V-SITL) environment. This proposed V-SITL environment offers increased scalability on account of the entire UAV system being simulated in a single computer. The TIPF mechanism has been simulated for 8 drones in a dynamic manner for square and triangle patterns. The simulation results show that the pattern formation time avoids collision in a time interleaving rate of 52.63%.**

***Index Terms*—Swarm of UAVs, Time Interleaved Pattern Formation, Centralised Coordinate Calculation, V-SITL, Square Pattern, Triangle Pattern**

## I. INTRODUCTION

From the past few years, Manned Aerial Vehicles (MAVs) had completely ruled the sky. With the development of Unmanned Aerial Vehicles (UAVs), a wide range of perks was readily available for aerial missions. UAVs have proved to be a significant asset to human society due to their ability to penetrate regions where humans cannot step their foot on. Research in UAV application and enhancements has brought about tremendous impact, bestowing these compact machines with unimaginable features. Nowadays, with the progression of technology, UAVs have been used without the integration of MAVs to facilitate higher levels of autonomy [5].

Multi-UAVs are deemed to be more powerful and useful under different scenarios like disaster relief, search, and rescue, etc. For such multi-UAV configuration, the primary setup consists of a Base Station (BS) and a Ground Control Station (GCS) that provides ground for wireless communication and centralized control of the multi-UAV architecture. One of the most commonly associated terms with UAVs is Swarm. A swarm is a group of coordinated drones that operate together to perform a mission. The success of the swarm missions depends on the inter UAV coordination with the help of the Micro Air Vehicle (MAV) Link protocol [14]. Unlike a single drone, a swarm of drones can achieve their goals more efficiently. This is because when compared to individual UAVs, swarms can effectively split a single large task among several drones, with each drone contributing a small portion of the entire work, thereby increasing the performance and modularity of the whole system. [12].

One of the most crucial swarm features is pattern formation. The drones are empowered with the capability of forming different patterns that adapt to the requirements of the underlying environment in which they are deployed [16]–[18]. This improves efficiency and reduces the time consumed for the entire mission. The current pattern formation mechanisms focus on fixing the number of UAVs, which helps in completely pre-planning the entire UAV mission [1]. However, such algorithms cannot be adapted to scenarios where the number of UAVs varies with every mission.

In addition, collision avoidance is a key feature that determines the success of any pattern formation mechanism. Traditional collision avoidance algorithms involve vision-based techniques, sensor-based techniques, etc. In these techniques, a lot of processing time is required resulting in an increased pattern formation time [19]–[22]. One another mechanism focuses on manually planning the trajectory of UAVs to avoid collisions. Although this is effective, it is tedious to implement and is not scalable as well. Thus, there is a need for a collision avoidance mechanism that is effective, scalable, easy to implement, and efficient in terms of pattern formation times.

For the pattern formation mechanism, there are multiple research findings with several algorithms. These developed algorithms can be directly deployed in actual UAVs and can be tested for effectiveness. Such testing is not efficient and can incur huge losses if the testing fails. Thus, there is always a compelling need for effective simulation and testing environment for a multi-UAV system that is scalable and cost-effective.

Thus, to mitigate the aforementioned issues in the conventional multi-UAV systems, the following solutions are proposed.

- A Time-Interleaved Pattern Formation (TIPF) mechanism facilitates collisionless pattern formation for a dynamic number of UAVs. This mechanism reduces the processing time compared to the existing systems.

- A Centralized Coordinate Calculation (CCC) algorithm helps to calculate the coordinates for dynamic number of UAVs under different patterns.
- A Virtual IP based Software In The Loop (V-SITL) setup is designed to effectively simulate and test the pattern formation algorithm among a swarm of UAVs.

The rest of the paper is organized as follows. The existing methodologies pertaining to multi UAVs and pattern formation are discussed in Section II. Section III gives a detailed insight into the proposed mechanism, the multi UAV setup, Hardware In The Loop (HITL), V-SITL, and TIPF. This section also discusses the mathematical formulation of CCC mechanism. Section IV presents the results of simulation for the proposed mechanisms and detailed performance analysis. Section V concludes the proposed mechanism, along with future work that can be undertaken with respect to pattern formation among multi UAVs.

## II. Related Work

The introduction of pattern formation in a multi-UAV system dramatically improves the efficiency of UAV missions. Automating the process of pattern formation reduces tedious human tasks, namely calculating UAV points, pre-defining trajectories to avoid the collision, etc [1]. In a multi-UAV system, the coordination and communication among the different UAVs are crucial for a successful mission which is achieved with the help of a lightweight protocol called MAVLink protocol [14]. MAVLink follows a modern hybrid publish-subscribe and point-to-point design pattern. The positioning of UAVs is crucial in the pattern formation mechanism. This is based on either maximizing the throughput or ease of network access [15]. The UAVs agree upon specific parameters like velocity, altitude, etc. and achieve the pattern formation based on such patterns of interest [1]–[4].

One of the efficient mechanisms for inter-UAV coordination is the leader-follower mechanism. In this mechanism, a single drone is selected as a leader and rest others as followers [12]. The leader broadcasts its positional coordinates and other state manipulation functions to all the followers based on which critical decisions are taken. The leader drone acts as a server, whereas all the followers act as clients [7].

The challenges of UAV in controlled formation environments include target classification, searching, and attack [8], [9]. These problems are mitigated by techniques like Particle Swarm Optimisation (PSO), Artificial Immune System (AIS), and Virtual Bee Algorithm (VBA) [13]. In remote environments, any absence of GPS signal significantly increases the difficulty in tracking the UAV coordinates. This issue is tackled by using techniques like bird flocking, laser finders, etc [10], [11].

This paper focuses on the TIPF mechanism for $n$ number of UAVs, employing a leader-follower mechanism, in a time-interleaved fashion. Also, it encompasses the CCC algorithm, which enables faster and easier coordinate calculations for a dynamic number of UAVs. The proposed TIPF model paves the way for scaling up the number of UAVs to as large as 1000 and also increases the flexibility of human testing by proposing a V-SITL configuration setup.

## III. Collisionless Time Interleaved Pattern Formation Mechanism

### A. Multi-UAV System Configuration

The multi-UAV setup consists of three major components, namely Base Station (BS), Ground Control Station (GCS), and the Remote Controlled (RC) Safety Trigger. Fig. 1 illustrates the environmental setup of multi-UAV configuration along with their nature of the interaction. The BS serves to provide a wireless network connection to the UAVs in remote areas where such connectivity would be scarce. UAV communication takes place in either 1.2 GHz or 2.4 GHz radio frequency channels. The frequency of 1.2 GHz allows drone communications up to 10 km, and this channel is licensed. However, the 2.4 GHz channel, is legal and typically provides a range of up to 1km.

The GCS is responsible for activities that include planning the UAVs' mission, coordinating the UAVs using the MAVLink protocol and continuously logging its positions and states. Thus, GCS instructs the UAVs to execute specific tasks, thereby forming the focal point of interaction between humans and UAVs. The RC Safety trigger is a device operated by humans that are used in anomalous or emergency situations. The RC trigger helps to perform safety maneuvers when the UAVs go out of GCS's control to prevent critical damage to the environment and the UAV.

### B. Hardware In The Loop (HITL) Configuration

The HITL setup provides a small scale environment where the drones are simulated using a computer network. Each computer representing a single drone is connected to a common router. The HITL configuration is set up in two different phases, namely UAV initialization and UAV communication establishment.

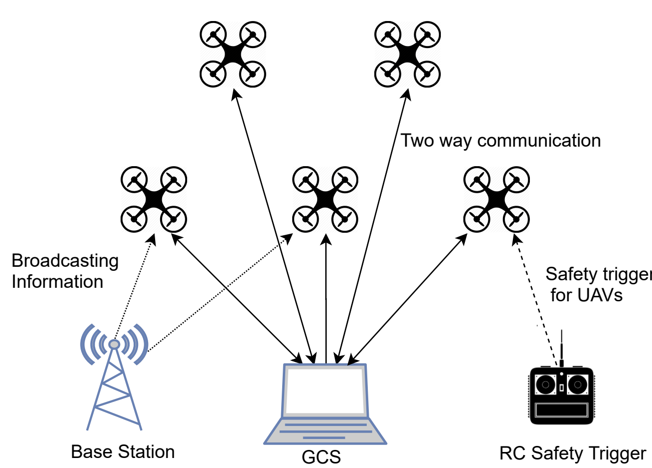


Fig. 1. Configuration Setup for Multi-UAV Environment

*1) UAV Initialization:* In this phase, the UAVs are initialized using the dronekit-SITL software package. The dronekit-SITL is used to create virtual drones that replicate the properties of the actual drones. The dronekit-SITL initializes the drones and returns a port address to which the MAVLink protocol should be connected.

*2) UAV Communication Establishment:* In this phase, the UAV communication is established using the MAVLink protocol. This protocol can be simulated using the python software package (mavproxy.py). This python package defines a set of ports in the UAV simulation environment, namely the master port and multiple output ports. The master port corresponds to the port provided by the dronekit-SITL software, whereas the output ports are used to communicate with the UAV and map them to the mission planner environment.

The steps to initialize and set up all the drones in the simulation environment is shown in algorithm 1. Although it is easy to configure, the HITL environment can support only a small number of drones as each drone requires an individual system for replication.

In a swarm environment, a large number of UAVs coordinate together to carry out a mission. The HITL setup is a tedious task due to the extensive requirement for a large number of UAVs in terms of computer systems. Hence, in the proposed TIPF model, the V-SITL configuration is used to solve the scalability issue.

### C. Virtual IP based Software In The Loop (V-SITL) Configuration

The V-SITL setup is similar in functionality when compared to the HITL setup, and uses the same phases as described above. In the V-SITL configuration, multiple UAVs are simulated using the same system. As every UAV has the same IP address, they can be differentiated using port numbers. However, every drone uses the same port numbers to log essential details like GPS, status, immediate command, and heading. Thus, this may lead to data incoherence as the data is written on the same port by every UAV. Hence, virtual IP addresses are essential to distinguish multiple drones.

---

**Algorithm 1** Hardware In The Loop Setup

---

**Input:** Number of drones $n_d$
**Output:** Multi UAV environment setup

**procedure** SETUPHARDWAREINTHELOOP($n_d$)
  Connect each system and the GCS to a common router
  Retrieve the IP address of each system using ifconfig as $systemIP$
  **for** $i = 1, 2, 3, ...n_d$ **do**
    Assign a port number as $outPort$
    Assign a port number in GCS as $gcsPort$
    Execute $dronekit - sitlcopter3.0 - Ii$
    Retrieve the port number as $masterPort$
    execute $mavproxy.py$ --$master$ $tcp$: $systemIP$: $masterPort$ --$sitl$ 127.0.0.1 : 5501 --$out$ $systemIP$:$systemPort$ --$mav$10 --$source$-$system$=1 --$out$ $gcsIP$ : $gcsPort$
  Assign a random drone as leader
  Assign the rest of the drones as followers

---

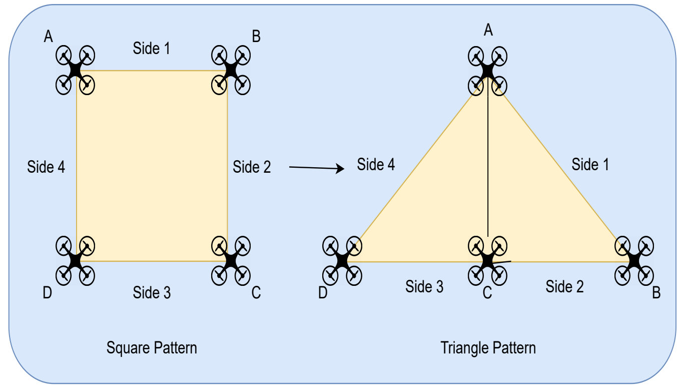


Fig. 2. Representation of Square and Triangle Patterns

---

**Algorithm 2** Virtual IP Creation

---

**Input:** The number of virtual IP addresses to be created $n_f$
**Output:** $n_f$ virtual IP addresses

**procedure** VIRTUALIPCREATION($n_f$)
  Switch to the root user
  Identify the interface of the current router using $ifconfig$ as $interfaceName$
  Identify the current IP address as $originalIP$
  Identify the current netmask address as $netmaskAddress$
  **for** $i = 1, 2, 3, ...n_d$ **do**
    $virtualIP = originalIP$ with last two digits offset to $i$
    execute $ifconfig$ $interfaceName$:$i$ $virtualIP$ $netmask$ $netmaskAddress$ $up$

---

The steps to create virtual IPs in the system are illustrated in algorithm 2. The number of virtual IP addresses to be created corresponds to one less than the total number of drones (followers) deployed in the environment with the source IP being assigned to the leader drone. As the number of simulation resources is independent of the number of UAVs, scalability to a large environment is easier.

### D. Time Interleaved Pattern Formation (TIPF)

The multi-UAV system follows a leader-follower pattern for communication [12]. A single drone is assigned as a leader, whereas all other drones act as followers. The leader drone coordinates the motion of all the drones with the help of GCS.

The proposed TIPF algorithm works for a dynamic number of drones ($n_d$). This algorithm forms numerous patterns like square and triangle, as shown in Fig. 2.

The TIPF mechanism uses the CCC algorithm to calculate the coordinates of each drone in a particular formation, like square and triangle. In the CCC algorithm, the leader is tasked with the responsibility of calculating the coordinates of the entire swarm system. This CCC algorithm is swift and easier to implement because the leader drone is well aware of the states of the entire multi-UAV system.

For each drone, the positional coordinates are represented as $(r, \theta)$, where $r$ represents the displacement of drone from the leader drone and $\theta$ represents the azimuth angle. The azimuth angle is the angle between the true north and the line joining the leader and the follower drone. The leader drone is assumed to be at the true north, and hence its coordinates are represented as $(0, 0)$. The calculation of points for each shape is pivotal, which is dynamically automated for any number of drones, $n_d$.

Fig. 2 includes the shapes, namely square and triangle, each containing 4 sides, and drones are equally divided among these sides. The four sides consist of the following point coordinates namely $(r_j, \theta_j, h_j)$ where $r_j$ represents the relative displacement from the leader drone, $\theta_j$ represents the azimuth angle and $h_j$ represents the altitude for the $j^{th}$ side. The drones on each side are characterised by $i_j$, denoting the $i^{th}$ drone of the $j^{th}$ side. The edge length, $l$, is calculated as the product of the highest number of drones in an edge and minimum inter-drone distance (10m). The number of drones per edge $n_{de}$ is calculated as,

$$n_{de} = \left\lfloor \frac{n_d}{4} \right\rfloor$$

*1) Square Pattern:* Each side of the square is represented by the following base equations,

**Algorithm 3** Square coordinate calculation

**Input:** Total number of drones $n_d$
**Output:** $n_d$ number of coordinates

**procedure** SQUARECOORDINATECALCULATION($n_f$)
  Remainder number of drones $r \leftarrow n_d \mod 4$
  **if** $r = 0$ **then**
    Edge Length $l \leftarrow n_{de} * 10$
    Use the base equations (1), (2), (3), (4) for point calculation
  **else**
    Edge Length $l \leftarrow (n_{de} + 1) * 10$
    **if** $r = 1$ **then**
      Use the base equation (2) for $side2$
      Increase range of $i2$ by 1
      In the other equations replace 10 by $\frac{l}{n_{de}}$
    **if** $r = 2$ **then**
      Use the base equation (2), (4) for $side2$ and $side4$ respectively
      Increase range of $i2$ and $i4$ by 1
      In the other equations replace 10 by $\frac{l}{n_{de}}$
    **if** $r = 3$ **then**
      Use the base equation (2), (3), (4) for $side2$, $side3$ and $side4$ respectively
      Increase range of $i2$, $i3$ and $i4$ by 1
      In the other equations replace 10 by $\frac{l}{n_{de}}$

$$r_1 = i_1 * 10, \theta_1 = 90° \tag{1}$$

$$r_2 = \sqrt{l^2 + {i_2}^2}, \theta_2 = 90° + \tan^{-1}(\frac{i_2 * 10}{l}) \tag{2}$$

$$r_3 = \sqrt{l^2 + (n_{de} - i_3)^2}, \theta_3 = 90° + \tan^{-1}(\frac{l}{(n_{de} - i_3) * 10}) \tag{3}$$

$$r_4 = (n_{de} - i_4) * 10, \theta_4 = 180° \tag{4}$$

$$\text{where} \quad 1 \le i_1, i_2, i_3 \le n_{de} \quad \text{and} \quad 1 \le i_4 < n_{de}$$

In the square formation, the azimuth angle is constant across all drones that are part of $side1$ and $side4$, having values of 90° and 180°, respectively. The hover height for all the drones is fixed as 20m irrespective of the side they belong to. The base equations represent the positional coordinates when the number of drones is an integral multiple of 4. The drones are assigned to each side in the order of $side_2$, $side_4$, $side_3$, $side_1$. The calculation of coordinates for square pattern formation for $n_d$ number of UAVs is illustrated in algorithm 3.

*2) Triangle Pattern:* Each side of the triangle is represented by the following base equations,

$$r_1 = i_1 * 10\sqrt{2}, \theta_1 = 135° \tag{5}$$

$$r_2 = \sqrt{l^2 + (n_{de} - i_2)^2}, \theta_2 = 180° - \tan^{-1}(\frac{n_{de} - i_2}{n_{de}}) \tag{6}$$

$$r_3 = \sqrt{l^2 + (n_{de} - i_3)^2}, \theta_3 = 180° + \tan^{-1}(\frac{i_3}{n_{de}}) \tag{7}$$

$$r_4 = (n_{de} - i_4) * 10\sqrt{2}, \theta_4 = 225° \tag{8}$$

$$\text{where} \quad 1 \le i_1, i_2, i_3 \le n_{de} \quad \text{and} \quad 1 \le i_4 < n_{de}$$

The triangle considered in this formation is an isosceles right-angled triangle at the vertex 'A'. Even though the triangle is a three-sided figure, we consider it containing four sides with hypotenuse being divided into two equal sides at 'C' as shown in fig. 2. The drones are assigned to each side in the order of $side_2$, $side_3$, $side_4$, $side_1$. The calculation of coordinates for triangle pattern formation for $n_d$ number of UAVs is elucidated in algorithm 4.

Once the coordinates are calculated for a shape, all the follower connections are checked to ensure the existence of the communication channel. The leader's position and heading are obtained to fix a destination position for the leader. Until the leader reaches the target position, it signals the follower drones to follow. All these processes have a time delay attached to it to maintain a synchronous motion between the leader and the followers.

**Algorithm 4** Triangle coordinate calculation

**Input:** Total number of drones $n_d$

**Output:** $n_d$ number of coordinates

**procedure** TRIANGLECOORDINATECALCULATION($n_f$)
    Remainder number of drones $r \leftarrow n_d \mod 4$
    **if** $r = 0$ **then**
        Edge Length $l \leftarrow n_{de} * 10$
        Use the base equations (5), (6), (7), (8) for point calculation
    **else**
        Edge Length $l \leftarrow (n_{de} + 1) * 10$
        **if** $r = 1$ **then**
            Use the base equation (6) for $side2$
            Increase range of $i2$ by 1
            In the other equations replace 10 by $\frac{l}{n_{de}}$
        **if** $r = 2$ **then**
            Use the base equation (6), (7) for $side2$ and $side3$ respectively
            Increase range of $i2$ and $i3$ by 1
            In the other equations replace 10 by $\frac{l}{n_{de}}$
        **if** $r = 3$ **then**
            Use the base equation (6), (7), (8) for $side2$, $side3$ and $side4$ respectively
            Increase range of $i2$, $i3$ and $i4$ by 1
            In the other equations replace 10 by $\frac{l}{n_{de}}$

Once the leader and the follower drones reach their destination, a braking command is issued to stop all the drones from moving ahead. This TIPF mechanism works well on a multitude of scenarios with different patterns and also results in a collisionless environment.

## IV. PERFORMANCE ANALYSIS AND RESULTS

The proposed TIPL algorithm is implemented successfully, and the data is plotted in a GUI framework supported by a Mission Planner tool. All the drones are connected to the Mission Planner using the port numbers that were initialized in the MAVProxy. Manual intervention is essential during the initial setup phase for establishing the connection between the UAVs and the Mission Planner environment. The administrator is provided with a complete track of the UAV Mission and its progress.

In the proposed TIPF model, the drones take off and rise to an altitude of 20m, then splits to form the square pattern, as shown in Fig. 3.

After the square formation, the drones fly forward for a very short distance before changing their formation to a triangle, as shown in Fig. 4. The drones then move for a short distance ahead, maintaining the triangle formation. Finally, when both the formations are completed, the drones are instructed to return back to their starting positions.

Fig. 5 shows the comparison of pattern formation time over an increasing number of drones. The proposed TIPF model outperforms the existing Vision-Based Pattern Formation (VBPF) [19] and Dynamic Programming based Pattern Formation (DPPF) models [22]. Due to the processing overheads associated with the VBPF mechanism, the pattern formation time increases exponentially with an increase in the number of drones. Although the DPPF mechanism has smaller pattern formation times for a lesser number of drones, it increases exponentially as the number of drones increases due to large memory overhead and increased data accession times. However, the TIPF mechanism has lesser processing and memory requirements compared to the VBPF and DPPF mechanisms, reducing the pattern formation time by 52.63%.

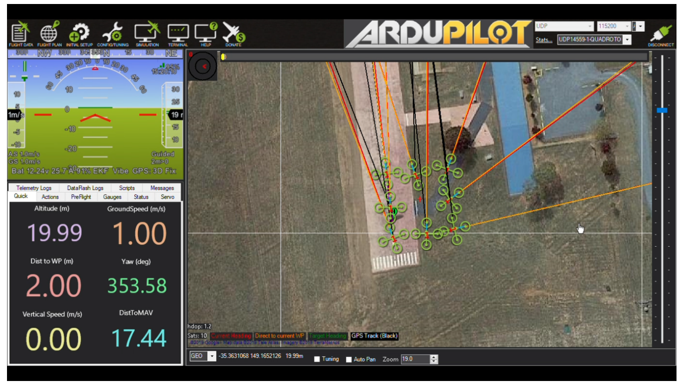


Fig. 3. Square Pattern Formation

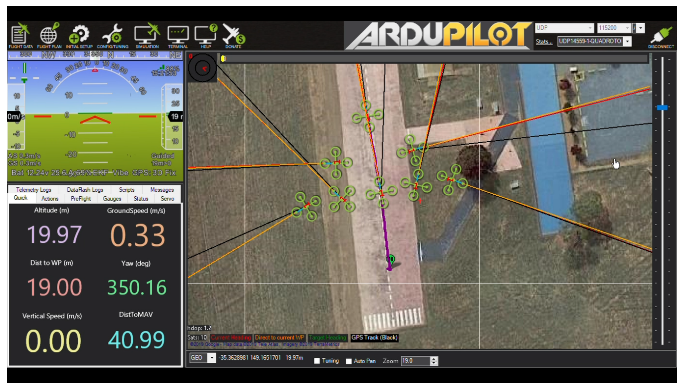


Fig. 4. Triangle Pattern Formation

Fig. 6 depicts the area covered by the multi UAV system for different patterns such as square and triangle in the CCC mechanism. It can be seen that the area covered increases quadratically when the number of drones increases. Further, for the same number of drones, the triangle pattern covers twice the area covered by the square pattern.

The coordinate calculation time is linearly proportional to the number of drones present in the system. Thus, the CCC mechanism has a lesser complexity of O(n) compared to the other existing mechanisms [1].

## V. CONCLUSION AND FUTURE WORK

The proposed TIPF algorithm offers collisionless pattern formation and also eases the process of coordinate calculation for dynamic number of UAVs. The V-SITL environment

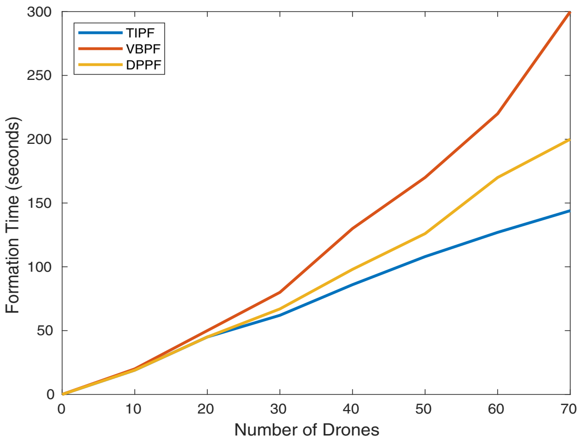


Fig. 5. Time taken for Pattern Formation in Different Models

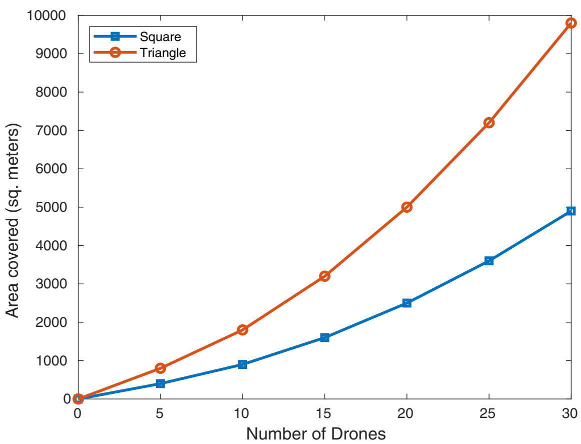


Fig. 6. Area Covered vs Number of Drones

provides increased scalability and testablity with lesser costs as only a single computer is sufficient to simulate the entire multi-UAV environment. In future, the concept of reinforcement learning and point clouds can be incorporated into pattern formation mechanism to improve automation and adaptability to a multitude of patterns.

## ACKNOWLEDGEMENT

This Publication is an outcome of the R&D work undertaken in the project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

## REFERENCES

[1] Ang KZY, Dong XX, Liu WQ, et al., "High-precision multi-UAV teaming for the first outdoor night show in Singapore," in *Unmanned Systems*, vol. 6, no. 1, pp. 39–65, 2018.

[2] X. Dong, B. Yu, Z. Shi and Y. Zhong, "Time-Varying Formation Control for Unmanned Aerial Vehicles: Theories and Applications," in *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 340-348, Jan. 2015.

[3] X. Wang, V. Yadav and S. N. Balakrishnan, "Cooperative UAV Formation Flying With Obstacle/Collision Avoidance," in *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 672-679, July 2007.

[4] S. Jacob et al., "A Novel Spectrum Sharing Scheme using Dynamic Long Short-Term Memory with CP-OFDMA in 5G Networks," in *IEEE Transactions on Cognitive Communications and Networking*, 2020

[5] J. Zhiqiang, Y. Peiyang, Z. Jieyong, Z. Yun and W. Xun, "MAV/UAV task coalition phased-formation method," in *Journal of Systems Engineering and Electronics*, vol. 30, no. 2, pp. 402-414, April 2019.

[6] S. Piao, Z. Ba, L. Su, D. Koutsonikolas, S. Li and K. Ren, "Automating CSI Measurement with UAVs: from Problem Formulation to Energy-Optimal Solution," *IEEE INFOCOM - IEEE Conference on Computer Communications, Paris, France, 2019*, pp. 2404-2412, 2019.

[7] F. Fabra, C. T. Calafate, J. C. Cano and P. Manzoni, "A methodology for measuring UAV-to-UAV communications performance," *14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, pp. 280-286, 2017.

[8] X. Zhang, H. Wang and H. Zhao, "An SDN framework for UAV backbone network towards knowledge centric networking," *IEEE INFOCOM - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Honolulu, HI, 2018, pp. 456-461, 2018.

[9] S. Martinez, "UAV Cooperative Decision and Control: Challenges and Practical Approaches (Shima, T. and Rasmussen, S.; 2008) [Bookshelf]," in *IEEE Control Systems Magazine*, vol. 30, no. 2, pp. 104-107, April 2010.

[10] J. Lwowski, A. Majumdar, P. Benavidez, J. J. Prevost and M. Jamshidi, "Bird Flocking Inspired Formation Control for Unmanned Aerial Vehicles Using Stereo Camera," in *IEEE Systems Journal*, vol. 13, no. 3, pp. 3580-3589, Sept. 2019.

[11] Jiadi Yang, Shuhang Zhang, Hongliang Zhang, and Lingyang Song, "Cooperative Trajectory Optimization for a Cellular Internet of UAVs," *ACM SIGCOMM - Association for Computing Machinery, New York, USA,* pp. 18–20, 2019

[12] W. Yuan, Q. Chen, Z. Hou and Y. Li, "Multi-UAVs formation flight control based on leader-follower pattern," *36th Chinese Control Conference (CCC)*, Dalian, pp. 1276-1281, 2017.

[13] Gunasekaran R., Uthariaraj V.R., Yamini U. et al, "A Distributed Mechanism for Handling of Adaptive/Intelligent Selfish Misbehaviour at MAC Layer in Mobile Ad Hoc Networks," *Journal of Computer Science and Technology*, vol. 24, pp. 472–481, 2009.

[14] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith and M. Khalgui, "Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey," in *IEEE Access*, vol. 7, pp. 87658-87680, 2019.

[15] G. Raja, A. Ganapathisubramaniyan, S. Anbalagan, S. B. M. Baskaran, K. Raja and A. K. Bashir, "Intelligent Reward based Data Offloading in Next Generation Vehicular Networks," *IEEE Internet of Things Journal*, 2020.

[16] R. Arshad, L. Lampe, H. ElSawy and M. J. Hossain, "Integrating UAVs into Existing Wireless Networks: A Stochastic Geometry Approach," IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, pp. 1-6, 2018.

[17] M. Aljehani and M. Inoue, "A swarm of computational clouds as multiple ground control stations of multi-UAV," *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, Nagoya, pp. 1-2, 2017.

[18] W. Chen, Y. Yaguchi, K. Naruse, Y. Watanobe, K. Nakamura and J. Ogawa, "A Study of Robotic Cooperation in Cloud Robotics: Architecture and Challenges," in *IEEE Access*, vol. 6, pp. 36662-36682, 2018.

[19] Wenjie Song, Yi Yang, Mengyin Fu, Fan Qiu, and Meiling Wang, "Real-Time Obstacles Detection and Status Classification for Collision Warning in a Vehicle Active Safety System", *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, March 2018.

[20] Sushil Pratap Bharati, Yuanwei Wu, Yao Sui, Curtis Padgett, and Guanghui Wang, "Real-Time Obstacle Detection and Tracking for Sense-and-Avoid Mechanism in UAVs", *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, June 2018.

[21] G. Raja, S. Anbalagan, V. S. Narayanan, S. Jayaram and A. Ganapathisubramaniyan, "Inter-UAV Collision Avoidance using Deep-Q-Learning in Flocking Environment," *IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York City, NY, USA, pp. 1089-1095, 2019.

[22] Qianru Wang, Bin Guo, Leye Wang, Tong Xin, He Du, Huihui Chen, and Zhiwen Yu, "CrowdWatch: Dynamic Sidewalk Obstacle Detection Using Mobile Crowd Sensing", *IEEE Internet of Things Journal*, Vol. 4, No. 6, December 2017