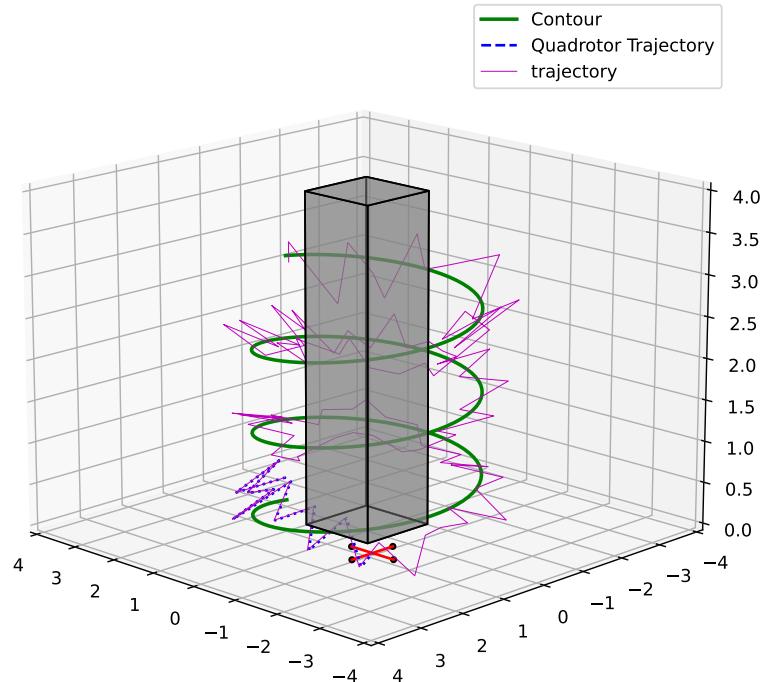


Graphical Abstract

Hyperchaotic Non-autonomous Discrete Memristor Map for 3D Boundary Surveillance by Quadrotor Robot

Harisankar R, Author Two, Author Three



Highlights

Hyperchaotic Non-autonomous Discrete Memristor Map for 3D Boundary Surveillance by Quadrotor Robot

Harisankar R, Author Two, Author Three

- Research highlight 1
- Research highlight 2

Hyperchaotic Non-autonomous Discrete Memristor Map for 3D Boundary Surveillance by Quadrotor Robot

Harisankar R^a, Author Two^b, Author Three^{a,b}

^a*Department One, Address One, City One, 00000, State One, Country One*

^b*Department Two, Address Two, City Two, 22222, State Two, Country Two*

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Keywords: Hyperchaotic map, Boundary surveillance, Quadrotor

PACS: 0000, 1111

2000 MSC: 0000, 1111

1. Introduction

Recently, quadrotors have become more popular for surveillance missions due to their unmatched manoeuvrability, ease of deployment, and reduced mechanical complexity compared to other aerial vehicles. Their compactness and agility make them very apt to fly in complex environments, thus suitable for open and confined spaces. Equipped with advanced sensors and cameras, quadrotors can capture high-resolution images and videos, thereby providing an all-detailed and comprehensive coverage of the area under surveillance. Additionally, the hovering ability along with the small accurate movements increases focus on areas of interest and improves the quality of surveillance data. Compared with conventional helicopters, quadrotors result in a rise in reliability and a significant reduction in the costs for manufacturing, operation and maintenance. Such characteristics outlined quadrotors more promising in a wide range of unmanned military and civilian missions,

like border patrol, search and rescue, and infrastructure monitoring. Similarly, in earlier works, researchers have designed various surveillance robots, leveraging advanced technology for inspection and monitoring using ground robots [1], intelligent obstacle avoidance through neural networks [2], and real-time monitoring systems in environmental and industrial applications [3]. Other applications of such robots encompass underwater surveillance [4], smart homes [5], power grid line inspection [6] and military surveillance utilizing unmanned aerial vehicles (UAVs) [7]etc.

The latest advances in quadrotor dynamics and control techniques are discussed, with the most recent developments and emerging trends found in those critical areas of UAV technology in [8]. For enhancing manual quadrotor operation, solving some of the critical challenges like precise trajectory tracking[9], vibration reduction and robustness against disturbances[10] are critical. Initially, the quadrotor creates thrust from the changing speeds of its four rotors, allowing it to control its movement and maneuver quite precisely in three dimensions. Later, equipped with sophisticated algorithms and sensors, navigation, obstacle avoidance, and real-time decision-making enabled a UAV to operate without human intervention. Progression into autonomous quadrotors in recently been given by the introduction of a kinodynamic path search and B-spline optimization for smooth, feasible, and time-efficient trajectories[11]. It drives quadrotors to fly in complex three-dimensional scenes with better precision and reliability, hence helping to increase their capability to decide in real-time and avoid obstacles in dynamic conditions. Later research flowed into adding other purposeful features in surveillance, specifically navigation [12, 13, 14] and localization even when in a swarm[15], object or location tracking [16]and more. Quadrotors use a monocular camera and minimal sensors to better the performance of ground target following in outdoor environments while trying to reduce system uncertainties and disturbances at the same time[16].

Surveillance missions can, in general, be divided into three categories: point, regional and boundary surveillance[17]. Among the various surveillance challenges, boundary surveillance which is mostly performed by humans can be efficiently handled by these robots discussed. Missions of surveillance can be categorized into point surveillance which regards a small and localized area or an object, regional surveillance which concerns larger areas, and boundary surveillance dealing with borders or perimeters. This paper focuses on the quadrotor-based boundary surveillance problem. Traditional surveillance methods generally rely on somewhat predictable patterns and are open

to detection and countermeasures. So the relevance of understanding and managing the unpredictability in flight path motion has become prominent. The unpredictability makes it hard for adversaries to foretell and avoid detection and ensures better coverage of the surveillance area. This can be realized with the divergence of close trajectories in chaotic maps[18].

There is a four-scroll chaotic attractor by Akgul et al [19], whose chaotic signals have wide applications in the majority of scientific and engineering fields. It forms very important signals in the design of random number generators, which are a basic requirement for any secure communication system since they offer robust encryption and protection of data. Also Chaotic signals could also be applied in image processing for secure image encryption to add security against access by unauthorized persons. And there is another paper [20], which provides sufficient contributions for 3D chaotic systems without equilibrium points to improve encryption techniques and secure communications, providing strong random number generation and video encryption solutions that fulfil international security requirements. This article by Zang [21] provides an overview of the chaos and fractal applications to robotics, showing their role in the improvement of mobile robot behaviour, optimization algorithms, bipedal locomotion, and modular robotic mechanisms.

In a recent article, Nwachioma et al.[22] presented a new chaotic system with four attractors, including fixed point and symmetric chaotic strange attractors, which are sensitive to initial conditions and show complex behaviour used as control input of a differential drive mobile robot to enable unpredictability and thorough scanning of the workspace. The space-filling maps of the chaotic system are used in creating paths that systematically cover the entire workspace and are quite utilized by the above robot. This is particularly valuable for tasks requiring thorough area coverage, such as search and rescue, surveillance, and autonomous exploration. Nasr et al.[23] contributed a multi-scroll chaotic system which improves efficiency in trajectory planning, maintains operational boundaries, integrates control with planning, and works with larger workspaces. The other type of application is that a multi-direction multi-scroll chaotic system model[24] will be presented along with its dynamics, FPGA implementation, and cryptographic application. Experimental results have shown that the proposed system can be very effective for generating highly sensitive chaotic sequences in robust image encryption. A chaotic system may be understood as one whose behaviour is highly divergent and, at the same time, seemingly random, even though

it is deterministic. All this is due to sensitivity to initial conditions, which ensure high security, making encrypted data very sensitive to the key and plaintext[24]. Additionally, the unpredictable behaviour of chaotic systems is utilised in trajectory planning for mobile robots to achieve complete and non-repetitive area coverage[22, 23]. Another relevance of chaotic trajectories in surveillance is that they evade forming any predictable pattern, making the mission of surveillance both stealthy and efficient.

In our application, we rely heavily on the property of unpredictability. While generating random points also gives unpredictability, there is a possibility that the same points or areas might get revisited repeatedly and also have the chance to happen consecutively. So, it cannot be used in applications that require independent trajectories. Chaotic trajectories, provide the required unpredictability while avoiding these consecutive overlaps. Also, apart from random prediction, chaotic points always fall within a chaotic pattern based on the system chosen. Further, chaotic trajectories are highly sensitive to initial conditions and system parameters, meaning small changes in these initial conditions can lead to vastly different behaviours. This parametric dependence is important because it provides an option to fine tune the trajectories to satisfy particular surveillance necessities, such as ensuring thorough coverage of critical areas or avoiding certain regions. And even a minute change in initial conditions is enough, if the target is to get a different trajectory which is mutually exclusive with the previous one. The capability to adjust parameters in real-time gives adaptability and flexibility to chaotic systems, which enables drones to respond effectively to changing conditions like moving obstacles or varying surveillance priorities.

The Algorithm proposed here [22, 23] offers a solution for path planning of robots in 2D applications only. However, this work addresses the boundary surveillance problem in three-dimensional space. Also, this is an extension of the earlier work done by P.S. Gohari[25], who utilized chaotic trajectories to enhance quadrotor performance in similar surveillance tasks. Some of the limitations of the above paper are: the definition of the closed contour was in a 2D plane, thus it constrained the quadroter vicinity within that closed contour. Since there is one technique that involves taking predefined points at the edges of real-world boundaries to form a definite contour. P.S. Gohari does not provide any mechanism to define the closed contour for surveillance. Specifically, it lacks a method for generating the minimal closed loop necessary to pass through these predefined points, which is necessary for collecting proper surveillance data. The approach by Gohari fails to account

for variations in elevation, and the absence of a vertical component within the trajectory reduces the system's adaptability to dynamic environments. And a 2D contour does not allow effective navigation around three-dimensional structures such as buildings, trees, or other variations in terrain. From this point of view, the quadrotor can miss important areas and details. so it may not deliver an effective surveillance application. This paper above the problem and improved quadrotor surveillance over the boundary in three-dimensional space by replacing the 2D closed contour[25] with a vertically extended open helical contour.

Another significant limitation of Gohari's paper is using the Henon map as a chaotic attractor to derive chaotic trajectories to fly quadrotors. The problem associated with the Henon map is that it creates chaotic attractor with limited occupation in the trapping region, resulting in a chaotic map with less spatial coverage. Details about trapping region and occupation of henon map within it is given in fig 2 of [25]. A hyperchaotic map was used to address this and improve the unpredictability while substantially increasing the spatial coverage provided by the Henon map. It is able to support significantly to generate mutually exclusive chaotic trajectories by using minute variations in the initial conditions in hyperchaotic systems, which ensures that more quadrotors can fly simultaneously on a single contour. Here are some applications showcasing the strength of hyperchaotic systems for better security and unpredictability in biometric authentication[26] and image encryption[27], leveraging their complexity in dynamics to improve resistance against various forms of attacks. P.S. Gohari mentioned in his paper[25] that the chosen henon system must be simple and efficient due to the limitations of quadrotor processors and power resources. However, our approach uses a hardware realizable hyperchaotic system that offers enhanced performance and efficiency for real-time applications than any other options. Further details are provided in the subsequent sections.

The main contributions are:

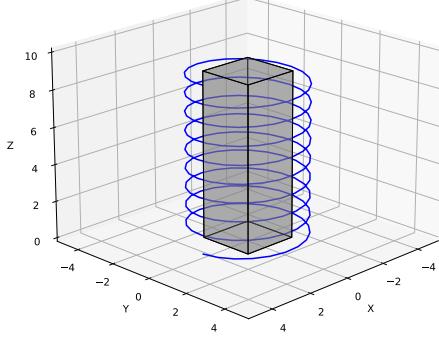
- The upgradation of the traditional 2D contour to a helical 3D contour in path planning greatly improves the effectiveness and unpredictability of the quadrotor's boundary surveillance, guaranteeing greater coverage, enhanced adaptability to navigate around vertical structures like buildings, and optimized monitoring capabilities in three-dimensional environments with a wider range of visual information.

- Hyperchaotic map instead of a Henon map like chaotic systems, significantly improving spatial coverage and the degree of unpredictability in the trajectory generated. Thus, it ensures much better coverage of the surveillance area and complex exploration, rendering effectiveness in real-time applications.
- The hardware realizable hyperchaotic system enhances its practical application in real-time quadrotor surveillance with cutting-edge performance and efficiency.
- The use of hyperchaotic systems allows for the generation of mutually exclusive chaotic trajectories through minute variations in initial conditions compared to a chaotic system, enabling multiple quadrotors to operate simultaneously on a single contour, significantly reducing the likelihood of overlap and interference between quadrotors operating on the same contour. Also, it provides high unpredictability due to sensitivity to the home position or initial condition of quadrotors.

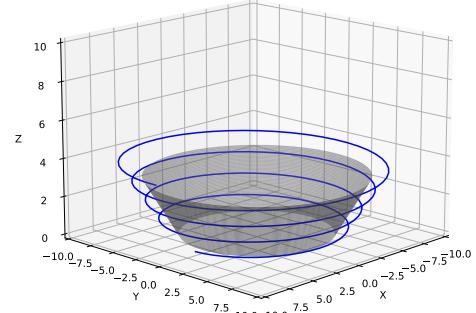
This paper is organized as follows. Section 2 describes the problem, which describes the usefulness of chaotic maps in surveillance and discusses the hyperchaotic attractor, together with bifurcation analysis. The methodology for generating 3D trajectories using hyperchaotic points is shown in Section 3. Section 4 presents results and discussions, comparing the Henon and Memester map's trajectories with an outline of the advantages of the Memester map. This section also includes the simulation of quadrotors along different hyperchaotic trajectories. Finally, Section 5 summarizes the main findings of the paper and offers some directions for future research.

2. Description of the problem

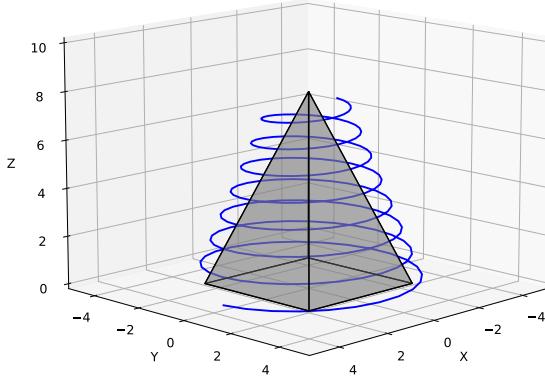
Chaotic maps play a key role in surveillance systems due to the properties that give invaluable advantages against traditional methods of random point generation as discussed earlier. Apart from the closed contour confined in the 2D plane by P.S Gohari[25], this paper introduces a helical contour that progresses around vertical structures or buildings based on a pitch value. The pitch refers to the distance in a vertical direction between successive turns of the helix and hence determines how tightly or loosely the contour wraps around the structure as it ascends.



(a) A cylindrical helical contour whose radius is constant.



(b) Helical contour whose radius is incrementing.



(c) Helical contour whose radius is decrementing.

Figure 1: Illustration of various open helical contours in three-dimensional space, which can be utilized for three structural navigation scenarios.

Figure 1 illustrates the concept of the helical contour progressing vertically upwards along the z-axis. Fig.1a. depicts a general or cylindrical helix whose radius is constant or the tangent of the helix makes a constant angle with a fixed line in space, and it is suitable for uniform ascension. And this open helical contour is surveilling around a vertical building. Additionally, Fig.1b contains a helix with an incrementing radius, which is particularly useful for navigating wide or expansive structures like stadiums. Conversely, Fig.1c is a helix with radius decrementing, ideal for structures like pyramids, where the contour naturally tightens as it ascends. The blue line represents an open helical contour in three-dimensional space, which is designated to be

monitored by the quadrotor. This contour serves boundary to the quadrotor to follow during ascending or descending around structures to ensure full coverage during surveillance.

In the domain of path planning for mobile robots, chaotic systems are mainly utilized in two ways, which are the generation of unpredictable trajectories and workspace coverage using chaotic-based optimization algorithms. This paper relies on the first way, which is using the inherent unpredictability of the system for generating trajectories that are quite hard to predict. In the surveillance problem discussed, a chaotic system generating unpredictable trajectories from two initial conditions with a difference of minute value would guarantee mutually exclusive paths. This means that these trajectories will never intersect each other, giving better coverage and security. The trajectory unpredictability ensures that the surveillance area is covered without repetition and with comprehensiveness. The coverage of chaotic paths over the entire surveillance region without leaving a significant gap is referred to as space-filling property. Nwachioma at[16] utilized this property to enable mobile for thorough scanning of the workspace, which is the other way of chaotic system utilization that was already mentioned.

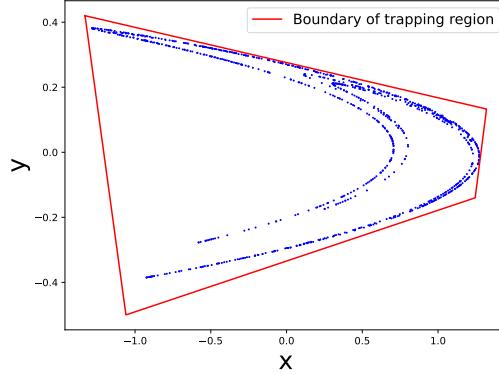
To enhance the degree of unpredictability in generated trajectories significantly, we adopted a hyperchaotic system building upon the work of chaotic trajectories in [20]. Along with enhancing unpredictability, the objective is to show how sensitivity to the initial conditions provides mutually exclusive trajectories for multiple quadrotors. And the comparison with henon map and detailed analysis of the hyperchaotic attractor are discussed below.

2.1. Description and comparative analysis of attractors

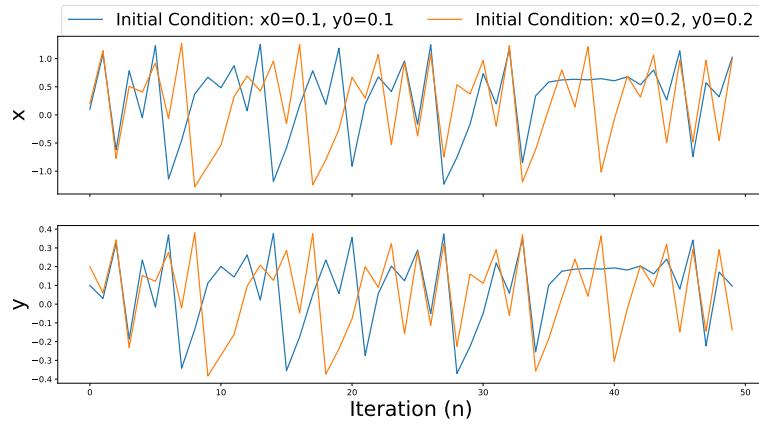
The initial stages of research focused on the Henon map to explore chaotic dynamics. Later, we introduced hyperchaotic systems to increase unpredictability, which would in turn yield multiple trajectories. To our knowledge, implementing hyperchaotic trajectories for multiple trajectories in real-world applications remains uncharted territory.

2.1.1. Henon map

This discrete-time dynamical system was introduced by Michel Henon in 1976 as a simplification of the Poincare section of the Lorenz system[37]. Despite its simplicity, the henon map is known to have chaotic behaviour due to the inclusion of a complex dynamical quadratic equation of x_{n+1}



(a) 2D plot of Henon map and its trapping region.



(b) x vs n and y vs n plot

Figure 2: Illustration of the trapping region and demonstration of sensitivity to initial conditions

below. Fig.2a shows the chaotic attractor of Henon map undergone 100000 iterations.

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{cases} \quad (1)$$

From the equations 1 above, for the specific values of the parameters where $a=1.4$ and $b=0.3$, the Henon map exhibits chaotic behaviour characterized by strange attractors. This sensitivity to initial conditions leads to aperiodic,

complex trajectories. Here trajectory is a sequence of the points (x_n, y_n) generated by iteration of the chaotic map's equations from an initial condition. This shows how the system evolves over time. As already mentioned, a minute difference in initial conditions is enough to produce multiple sequences of points, which are non-repetitive and complex. ie mutually exclusive paths. In Fig.2b, it can be seen that incrementing the initial conditions by 0.1 gives divergent trajectories.

The chaos in the Henon map can be evaluated using Lyapunov exponents, which measure the rate at which nearby sequence of points diverge. Since the Henon map is two-dimensional, it has two Lyapunov exponents. The largest Lyapunov exponent having a positive value, $\lambda_1 = 0.418$, indicates chaotic behaviour, and the negative second exponent, $\lambda_2 = -1.622$, in the orthogonal direction, confirms the system's chaos. The trapping region of the Henon map represents the area where the sequence of points remains bounded and does not escape once they enter from outside. It is bounded by a quadrilateral ABCD whose vertices are A = (-1.33, 0.42), B = (1.32, 0.133), C = (1.245, -0.14), D = (-1.06, -0.5) [37]. The trapping region is marked as a red line in Fig.2a.

2.1.2. Other chaotic systems

Depending on the purpose, other chaotic systems can also be utilized as implemented in [20]. One of the important chaotic maps is the standard circle map mentioned in [38]. One merit of this system is that it is capable of generating points equidistant from the centre, which is not possible with the Henon map described earlier. This feature is useful in cases when there is a need to choose a proximity around the contour within a certain radius. Similarly, trapping the generated points within a certain radius is discussed in [20], where the Bogdanov map undergoes an affine transformation with radius parameter d.

2.1.3. Quadratic discrete memristor map

To achieve a higher degree of unpredictability and rate of divergence of the nearby sequence of points, the chaotic theory proposes the replacement of the current chaotic system with a hyperchaotic one. The arrival to the discrete memristor map happened because it can be implemented in digital circuits, which improves hardware efficiency, flexibility and latency for real-time processing. Also this map can be realized using realistic devices in analog[18].

In general, the existence of hyperchaos requires at least four dimensions in continuous dynamical systems[13-14], while for discrete dynamical systems, it is possible to obtain it with two dimensions[15-16]. In particular, continuous memristor-based systems in the (ψ, q) domain cannot reach hyperchaos because they only have three dimensions[17]. but they can be hyperchaotic in the (v, i) domain with four dimensions. However, 2D memristor-based discrete maps are capable of generating hyperchaos[16,18,19]. This makes discrete systems advantageous over continuous ones in terms of less complex algebraic equations [20]and higher computational efficiency, which can be used for chaos-based industrial applications[21].

Initially, a discrete memristor model was established with the use of the forward Euler difference method [18], where a novel 2D hyperchaotic map was developed from a memristor-capacitor circuit. Houzhen Li et al. [19] introduced a discrete charge-controlled memristor with cosine memductance and developed a generally structured method for the creation of 2D memristive maps by coupling this discrete memristor with various 1D discrete maps. Han Bao and Houzhen Li et. al[ref] utilized this framework to produce four 2D DM(discrete memristor) maps employing different memristance nonlinearities in terms of unified mapping model. The theory associated with producing a 2D hyperchaotic memristive map is as follows.

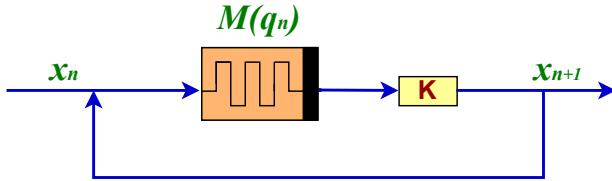


Figure 3: Schematic of unified DM mapping model

From the circuit theory perspective, a memristor is a two-terminal nonlinear device that shows a pinched hysteretic loop in the voltage-current plane under periodic input by voltage $v(t)$ or current $i(t)$. An ideal charge-controlled memristor[9] is given by.

$$\begin{cases} v(t) = M(q) \cdot i(t) \\ \frac{dq(t)}{dt} = i(t) \end{cases} \quad (2)$$

where q is used to denote the variable for charge, and $M(q)$ denotes the

memristance in Ohms.

The continuous memristor model can be discretized using the forward Euler difference method[18,19]. Let (v_n, i_n, q_n) be the sampled values of voltage $v(t)$, current $i(t)$, and charge $q(t)$ at the n -th iteration and q_{n+1} is the sampling value at the $(n + 1)$ -th iteration. An ideal discrete memristor (DM) can then be modelled by

$$\begin{cases} v_n = M(q_n) \cdot i_n \\ q_{n+1} = q_n + i_n \end{cases} \quad (3)$$

where $M(q_n)$ represents the sampling value of memristance $M(q)$ at the n -th iteration.

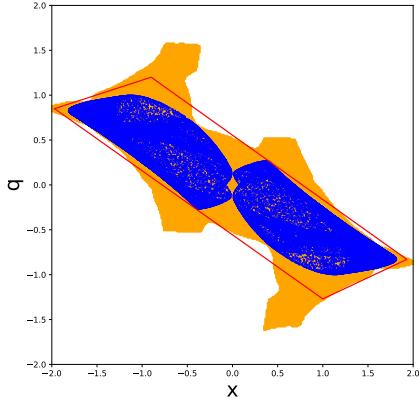
Han Bao et. al[ref] proposed four discrete memristors with different non-linear memductances according to the mathematical expressions of continuous memristors[1,32,33]. Numerical analysis proved that these DMs inherit the fundamental properties of continuous-time memristors[1]. These DMs were further utilized to generate four types of discrete hyperchaotic maps by adopting it with a unified mapping model. The schematic of the unified DM mapping model is shown in Fig.3, which employs a simple scaling via a proportional controller k . Through this approach, Han Bao et al. [ref] successfully generated four 2D DM hyperchaotic maps. The mathematical equations for a unified DM mapping model can then be expressed as

$$\begin{cases} x_{n+1} = kM(q_n)x_n, \\ q_{n+1} = q_n + x_n \end{cases} \quad (4)$$

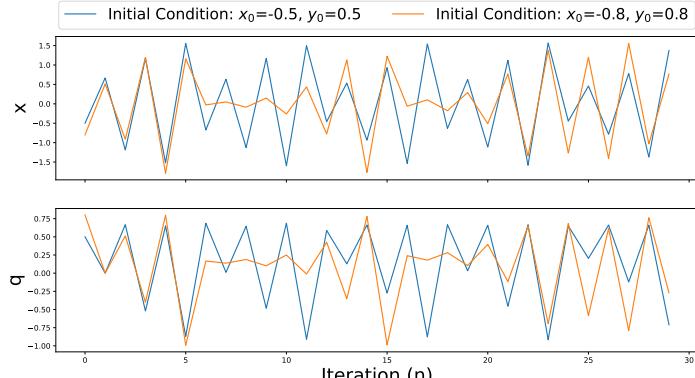
where k is the control parameter. Among the four DMs taken in [ref], Quadratic DM (Q-DM) provided a slightly higher LE_1 (Lyapunov exponents- (LE_1, LE_2)) than other models, pointing out its hyperchaotic complexity. Consequently, this paper uses the Q-DM map for generating chaotic trajectories. Thus, the sampling value of memristance $M(q_n) = q_n^2 - 1$ for quadratic discrete memristor is applied to obtain the corresponding 2D Q-DM hyperchaotic map equations Eq.5.

$$\begin{cases} x_{n+1} = k(q_n^2 - 1)x_n, \\ q_{n+1} = q_n + x_n \end{cases} \quad (5)$$

Fig.4a illustrates the 2D Hyperchaotic Q-DM map. High hyperchaotic complexity is achieved with control parameter $k = 1.78$ and initial condition



(a) 2D plot of Q-DM map and its trapping region.



(b) x vs n and y vs n plot

Figure 4: Illustration of the trapping region and demonstration of sensitivity to initial conditions of Q-DM map

$(x_0, q_0) = (-0.5, 0.5)$ as in [ref]. Additionally, the corresponding graphs of x versus n and q versus n for slightly different initial conditions are presented in Fig.4b, showing the dynamic behaviour of the system for iterations.

This paper used a method for finding the trapping region of the Q-DM attractor by checking whether the output from all iterations for given initial conditions within the range x_0 from -2 to 2 and q_0 from -2 to 2 remains within a defined quadrilateral around the Q-DM map. In Fig.4a, the red

line indicates the predefined quadrilateral and the orange region indicates the trapping region.

2.1.4. 2D non-autonomous discrete memristor-based hyperchaotic map

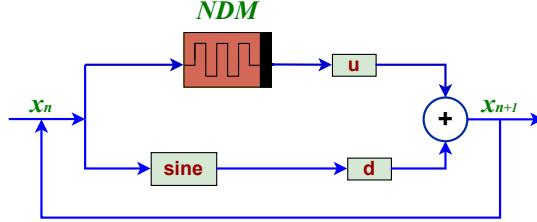


Figure 5: Schematic of framework for NDMH map

In the Q-DM map, all the chaotic points generated lie in the Q-DM attractor and follow a certain inclination within a certain range as in Fig4a. It was observed that in the process of tuning at $k=1.78$, two attractors almost touched each other. However, further variations in the parameter k do not merge these attractors. Furthermore, the space-filling attribute of attractors in the vicinity of the origin $(0,0)$ is very low, indicating a low density of trajectories in that region. So, the probability of hyperchaotic points occurring is lower in that region. The inclination of the attractor is perceptible in the red quadrilateral of the Q-DM map defined in Fig4a, where all the chaotic points reside in a certain band with respect to the x -axis. This is in contrast to the Henon map, where the trapping boundary lacks any discernible inclination illustrated in Fig2a.

To enhance hyperchaotic behaviour, a new 2D nonautonomous discrete memristor-based hyperchaotic map was presented in recent research[ref2] by Mengjiao Wang. It combines the new non-autonomous discrete memristor with the sine map and provides a new 2D NDM-based hyperchaotic map. Inspired by [45], Mengjiao Wang et al. [ref] proposed a new memristance function $M(q_n) = q_n^2 - 1 + c \cdot \sin(r \cdot n + r)$ in the ideal discrete memristor Eq.2. Consequently, the new NDM mathematical model can be expressed as follows

$$\begin{cases} v_n = [q_n^2 - 1 + c \cdot \sin(r \cdot n + r)] \cdot i_n \\ q_{n+1} = q_n + k \cdot i_n. \end{cases} \quad (6)$$

A new NDMH 2D map is built by using the above NDM. A schematic diagram for the framework of the NDMH map is shown in Fig.5. The mathematical model of a new 2D NDMH map is:

$$\begin{cases} x_{n+1} = u \cdot x_n \cdot [q_n^2 - 1 + c \cdot \sin(r \cdot n + r)] + d \cdot \sin(\pi \cdot x_n) \\ q_{n+1} = q_n + k \cdot x_n \end{cases} \quad (7)$$

where c, k, u, r , and d are control parameters. When the parameters are set to $c = 0.56$, $u = 0.8$, $k = 1$, $r = 0.05$, $d = -0.8$, and the initial value is $(x_1, q_1) = (0.1, 0.1)$ as per [ref], the hyperchaotic attractor of the NDMH map is shown in Fig..

2.2. Kinematic relative motion

In this paper, analysis is done in a moving frame where the quadrotor is considered stationary. To generate hyperchaotic trajectories for surveillance purposes, the 2D hyperchaotic map needs to be plotted with respect to the moving frame, ensuring that the hyperchaotic behaviour is accurately represented in that frame. Here in Fig.6a, the Q-DM hyperchaotic 2D map is plotted on the x-y plane of the moving frame, where the quadrotor is stationary. The blue line in Fig.6a indicates the 3D closed elliptical contour, which is the path followed by the quadrotor. For that, our objective is to conduct an affine transformation, which involves the transformation of a point from a global stationary frame F' to a moving frame F . An affine transformation is a means of linear mapping that preserves points, straight lines, and planes but not angles and Euclidean distances.

The theory of Affine transformation is as follows:

Consider a three-dimensional domain with a stationary coordinate system \mathcal{F}' as shown in Fig. 6a. In this coordinate system, a closed 3D contour C , which is an ellipse, is defined. The origin O of a translating coordinate system \mathcal{F} slides on this contour and performs a periodic motion. The relative motion of this system is described as:

$$\vec{r}'_s[n] = R[n] \vec{r}_{s/o}[n] + \vec{r}'_o[n] \quad (8)$$

where $\vec{r}'_s[n]$ and $\vec{r}_{s/o}[n]$ are the position vectors of a trajectory point in the fixed and moving frames at discrete time step n , and $\vec{r}'_o[n]$ is the position vector of the origin O of the moving frame with respect to the fixed frame

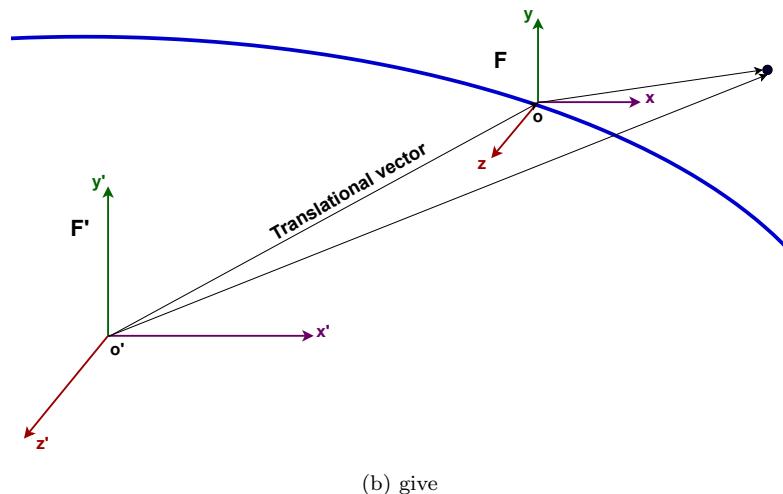
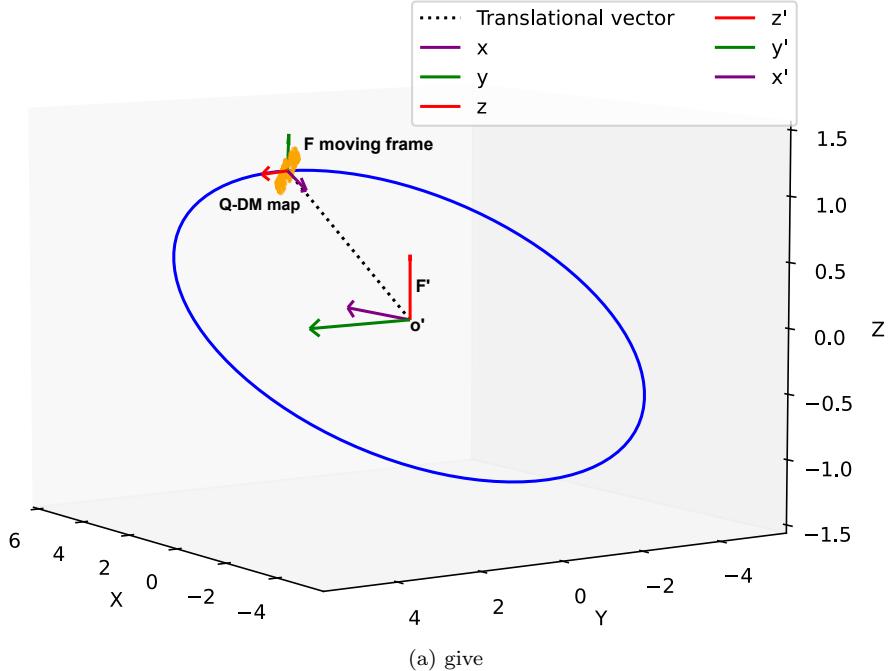


Figure 6: give

at the same discrete time step. The rotation matrix $R[n]$ rotates the moving frame \mathcal{F} to the fixed frame \mathcal{F}' at discrete time step n . It should be noted that in the paper, the parameters with a prime ' refer to those defined with respect to the fixed coordinate frame, while the others are defined with respect to

the moving frame.

For expressing the relative motion equation Eq.8 in an affine matrix form, we use the rotation matrix and the translational vector as a matrix. This provides the ability to express the transformation between coordinate systems in a unified matrix format and is also efficient in implementing in firmware. For that, we describe the coordinates, (x, y, z) of a point with respect to the moving frame F and given translational vector \mathbf{t}_n , which describes the origin of the moving frame with respect to the fixed frame. The coordinates (x', y', z') of the point with respect to fixed frame F' can be determined in affine form as follows:

$$\begin{bmatrix} x'_n \\ y'_n \\ z'_n \\ 1 \end{bmatrix} = \begin{bmatrix} [R]_n & [\mathbf{t}_n]^T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} \quad (9)$$

The rotation matrix is first applied to align the coordinates from the moving frame to the global frame during the transformation process and expressed in Eq.10

$$([R]_{ij})_n = \mathbf{e}'_i \cdot \mathbf{e}_j, \quad i, j = 1, 2, 3 \quad (10)$$

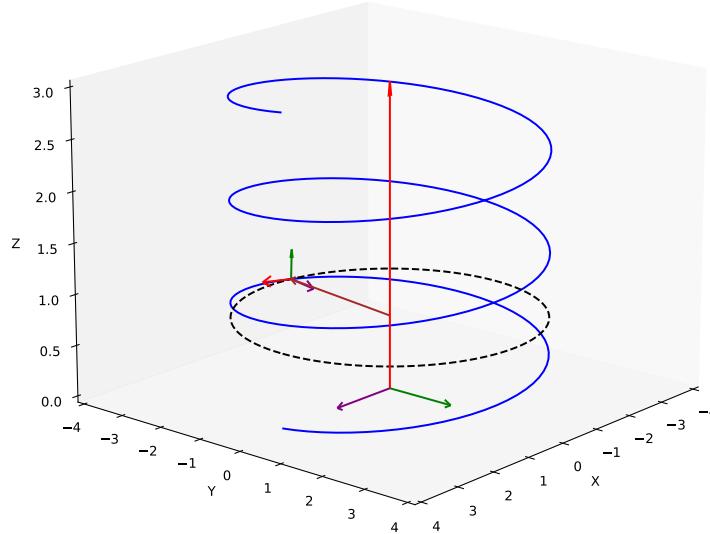
The application of the Euclidean affine transformation matrix for transforming hyperchaotic points from a fixed frame into a moving frame ensures that the hyperchaotic properties of the system are preserved. This is due to the fact that this affine transformation defined by Eq.9 does not alter the essential characteristics of the chaotic systems, such as boundedness, determinism, and the divergence of nearby trajectories[39]. In other words, this intrinsic hyperchaotic nature of the points remains unchanged even after affine transformation.

3. Methodology

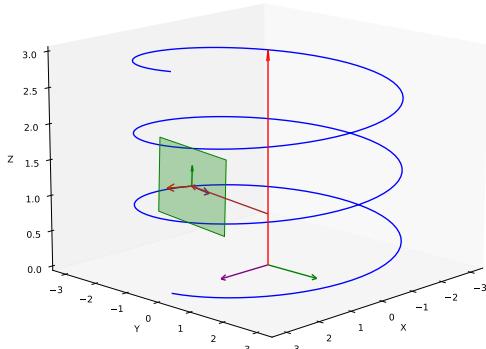
The abstract of the proposed algorithm or method is as follows:

A plane holding a two-dimensional hyperchaotic Q-DM map slides along an open helical contour projected vertically upward around a building and acts as a guiding line for a quadrotor. The normal vector of the plane is kept parallel to the tangential vector of the contour at every discrete point. After generating a sequence of points on successive planes, connecting them with a curve gives the desired trajectory.

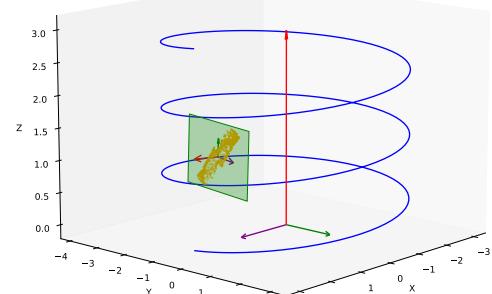
— Tangential unit vector or \hat{z}_n	— Radius vector
— Normal unit vector or \hat{y}_n	- - - Circle of Radius
— \hat{x}_n	



(a) give



(b) give



(c) give

Figure 7: give

This paper presents an open helical contour that advanced incrementally around vertical structures like buildings according to a pitch value. And pitch is defined as the distance between two consecutive turns in a helix. Parametric equations are used to plot the contour of the helix, using parameters like initial radius r , radius growth rate g , and pitch as p . The initial radius

sets the starting size of the helix, while the radius growth rate describes how the radius may increase as in Fig.1b or decrease as in Fig.1c along the helix. And if the radius growth rate is 0, then the contour will be the cylindrical helix as in Fig.1a. These parameters are used as follows to define the contour of the helix.

$$\begin{cases} x'(t) = (r + gt) \cos(t) \\ y'(t) = (r + gt) \sin(t) \\ z'(t) = \frac{pt}{2\pi} \end{cases} \quad (11)$$

To determine the tangent vectors along the helical contour, the first derivatives of the parametric equations with respect to t are taken. The derivatives below give the components of the tangent vector at any point along the helix.

$$\begin{cases} \frac{dx'(t)}{dt} = -(r + gt) \sin(t) + g \cos(t)t \\ \frac{dy'(t)}{dt} = (r + gt) \cos(t) + gt \sin(t) \\ \frac{dz'(t)}{dt} = \frac{p}{2\pi} \end{cases} \quad (12)$$

The contour is then discretized to account for accurate computations and simulations. This discretization is necessary to adapt the kinematic relative motion equations Eq.8 for an n -th point on the helical contour. And a parameter called *steps* is added to the algorithm to generate a specified number of equidistant points along the open helical contour. Then, the next goal is to plot moving frames at those discrete points. Fig.7a depicts the n^{th} point on the general helical contour, represented by coordinates (x'_n, y'_n, z'_n) , and a radius vector is drawn to a point $(0, 0, z'_n)$ on the global z' axis from this particular point, which shown as a brown line in Fig.7a. This radius vector corresponds to the circle passing through the n^{th} point on the contour, as depicted by the dotted black line. In this figure, the global z -axis has been extended for clarity to show where the radius vector actually connects. And it is considered that the quadrotor moves from the bottom toward the top in an anticlockwise direction. So the direction of the tangent follows this motion, as illustrated in Fig.7a. The normal vector is obtained by taking the cross product of the tangent vector and the radius vector, as given below.

$$N(n) = \vec{t}(n) \times \vec{r}(n) \quad (13)$$

Where $\vec{t}(n)$ is calculated based on the Eq.12 above, at an n th point on contour and $\vec{r}(n)$ is the radius vector from the n th point on contour. Note

that Eq.13 is an adapted equation due to the direction of the radius vector taken from the nth point on the contour to global z' axis.

The unit vectors of the normal vector, radius vector and tangent vector are denoted as \hat{y}_n , \hat{x}_n , and \hat{z}_n respectively, which represent the axes of moving frame at the n th point on the contour is shown in Fig.7a. For an arbitrary contour curve, which is not the defined open helical contour, the above mentioned method for finding the moving coordinate system is not valid. As pointed out by P.S. Gohari et al.[20] in these instances, \vec{z}_n is considered to be the vector pointing towards the next point in the contour, and \vec{z}_0 is the position vector from the origin of the global frame to the first point in the contour. Then \hat{z}_n , \hat{y}_n , and \hat{x}_n are written as.

$$\begin{cases} \hat{z}_n &= \frac{\vec{p}'_{n+1} - \vec{p}'_n}{\|\vec{p}'_{n+1} - \vec{p}'_n\|}, \\ \hat{y}_n &= \frac{\hat{z}_0 \times \hat{z}_n}{\|\hat{z}_0 \times \hat{z}_n\|}, \\ \hat{x}_n &= \hat{y}_n \times \hat{z}_n, \end{cases} \quad (14)$$

where \vec{p}'_n is the position vector of the n th point on the contour. In the above case, one constraint is that when part of the guiding line is almost parallel to the reference unit vector \hat{z}_0 , the \hat{y}_n in that point cannot be defined since $\hat{z}_0 \times \hat{z}_n = 0$. The y -axis in such a case can be selected though as $\hat{y}_n = \frac{\hat{z}_{n-1} \times \hat{z}_n}{\|\hat{z}_{n-1} \times \hat{z}_n\|}$. However, in the proposed method, this constraint does not apply, as the radius vector is used to plot the normal vector. Because the radius vector will always be perpendicular to the tangential vector.

The kinematic relative motion Eq.8 is used for transforming points in 3D coordinates. However, our approach requires transformation only in two dimensions. So the Eq.8 is reduced as below for that purpose.

$$R' = \vec{p}'_n + x * \vec{r}(n) + y * (\vec{r}(n) \times \vec{t}(n)) \quad (15)$$

where R' vector represents the position vector of a point and \vec{p}'_n is the position vector of the n th point on the contour with respect to the global frame. This equation is meant to plot a point in the 2D xy plane of the moving frame relative to the global frame. Through this reduced affine transformation Eq.15, any point can be plotted on any n th moving frame while preserving its geometric affinity. Strogatz et. al[39] validated that kinematic relative motion in Eq.8 is affine. By sequentially applying the hyperchaotic points through this Eq.15, the 2D Q-DM map is effectively plotted on the n th moving frame along the contour. Prior to this, Eq.15 is utilized to show xy

planes of a specific width by taking coordinates of vertices and visualised in Fig.7b. Later, the 2D hyperchaotic Q-DM map is plotted on the nth moving frame along the contour, which is shown in Fig.7c. A desired trajectory is obtained by mapping a sequence of points on successive planes, connecting them with a smooth or linear curve.

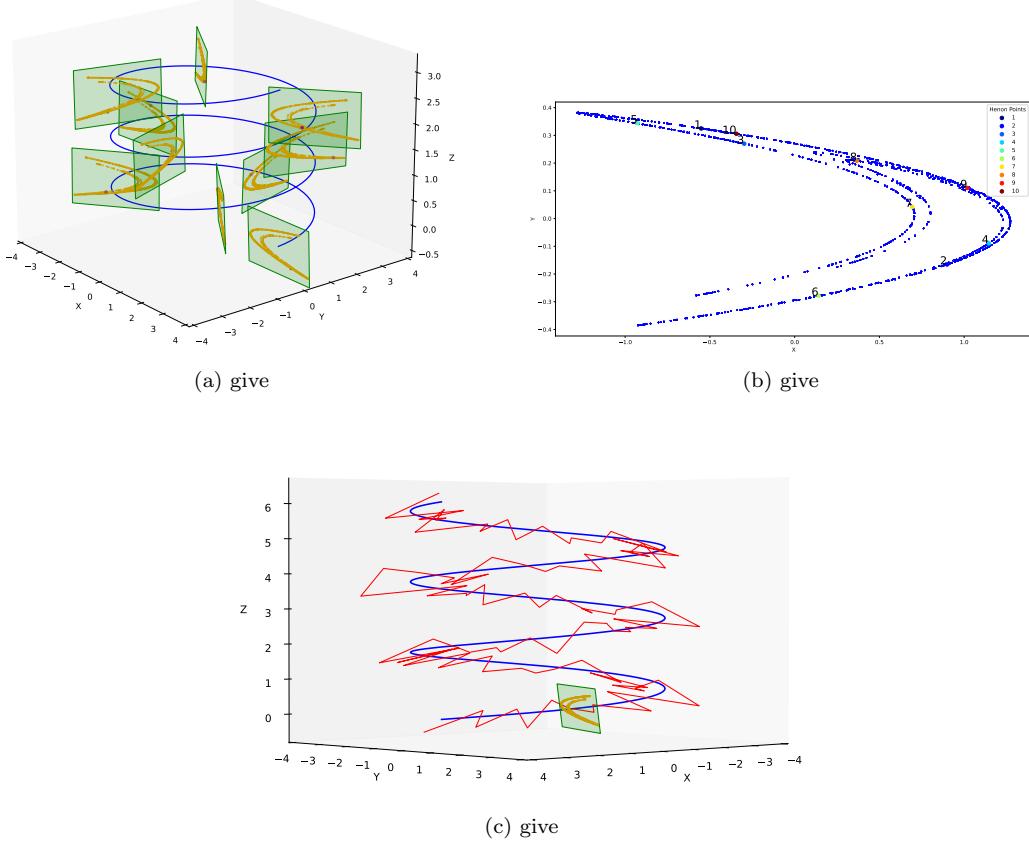


Figure 8: give

The verification of whether the sequence of points are part of a chaotic map was conducted through visualization. For that, the Henon map was plotted on all xy planes of the moving frames, and the sequence of points was generated in each plane, as in shown Fig.8a as purple dots. The number of planes is 10, as determined by the parameter 'steps', which is set to 10. The Henon map was chosen for a more thorough analysis during visual validation since it is a narrow attractor and has less coverage in its 2D plot. Then the

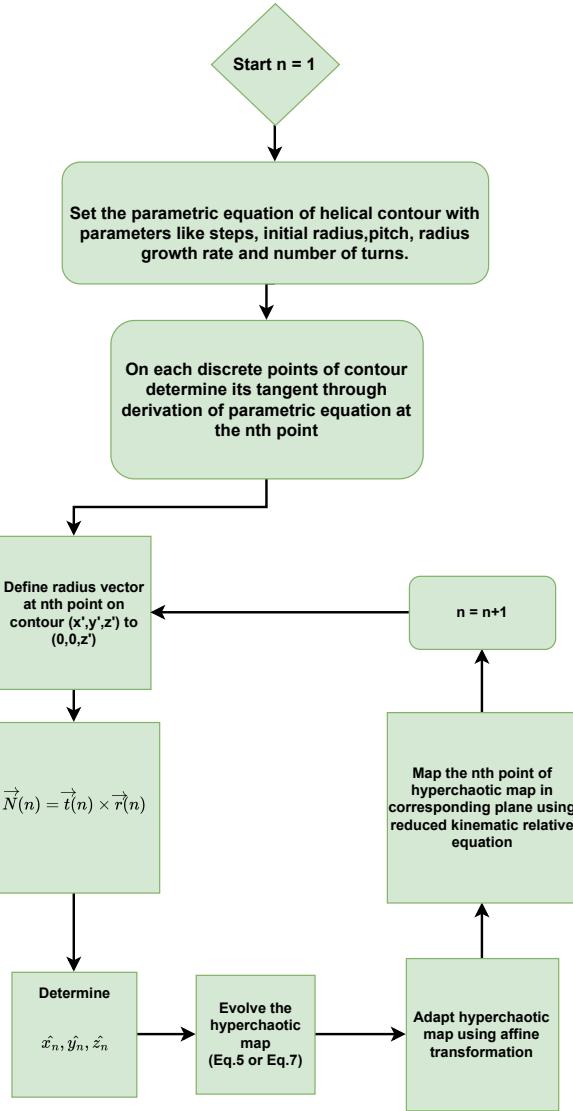


Figure 9: Flowchart of proposed algorithm

position coordinates of these sequences of points on all planes were subjected to a reverse transformation of Eq.15 to a common plane as shown in Fig.8b. And different colours are assigned to each point extracted from the planes. In this way, it is possible to confirm if they were part of a chaotic attractor or not. Fig.7c shows a linear trajectory plotted through a sequence of points

in the iteration range 49,000 to 50,000. To enhance the visualization and show more clearly the unpredictability of the trajectory, the simulation set the parameter 'steps' to 80 and pitch as 2 with a helix of 3 turns. Fig.9 shows the flowchart of proposed method.

4. Quadrotor simulation model

The quadrotor is a four-rotor UAV. It works based on a set of ruling equations that describe the motion and control of this quadrotor [42]. Kinematically, the quadrotor's placement in 3D space is modelled by a position vector $\mathbf{p} = [x, y, z]^T$, while the orientation is modelled by roll (ϕ), pitch (θ), and yaw (ψ). The rotation matrix \mathbf{R} transforms the forces and torques from the body frame into the inertial frame is expressed as:

$$\mathbf{R} = \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\theta) \cdot \mathbf{R}_x(\phi) \quad (16)$$

Roll (ϕ). : The rotation about the X -axis is given by:

$$\mathbf{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (17)$$

Pitch (θ). : The rotation about the Y -axis is given by:

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (18)$$

Yaw (ψ). : The rotation about the Z -axis is given by:

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (19)$$

where $\mathbf{R}_x(\phi)$, $\mathbf{R}_y(\theta)$, and $\mathbf{R}_z(\psi)$, are the basic rotation matrices for roll, pitch, and yaw respectively. An equal adjustment in the speed of the diagonal motors causes a change in the yaw angle, as illustrated in Fig. 8(a). Conversely, if the rotational speed of one rotor on the roll/pitch axis is altered relative to the corresponding rotor on the same axis, it results in roll or

pitch motion, as shown in Fig. 8(b) and Fig. 8(d). The dynamics associated with translation are due to the thrust force expressed by:

$$\mathbf{F} = m \cdot (g + \mathbf{a}) \quad (20)$$

In that context, m stands for the mass of the quadrotor, g is the acceleration of gravity, and \mathbf{a} is the acceleration that would be desired. Specifically, thrust force can be written as:

$$\mathbf{F} = m \cdot (g + \text{des_z_acc} + K_{p_z} \cdot (\text{des_z_pos} - z) + K_{d_z} \cdot (\text{des_z_vel} - \dot{z})) \quad (21)$$

where des_z_acc is the desired vertical acceleration, des_z_pos is the desired vertical position, des_z_vel is the desired vertical velocity and K_{p_z} and K_{d_z} are the proportional and derivative gains, respectively. The rotation dynamics are given by:

$$\mathbf{T} = \mathbf{I} \cdot \boldsymbol{\alpha} + \boldsymbol{\omega} \times (\mathbf{I} \cdot \boldsymbol{\omega}) \quad (22)$$

where \mathbf{I} is the moment of inertia matrix, $\boldsymbol{\alpha}$ is the angular acceleration, and $\boldsymbol{\omega}$ is the angular velocity vector. Control inputs are applied through thrust control, a method that works on the vertical movement of the object as in Eq.21 and torque control, which controls the roll, pitch, and yaw of the object in Eqns.17, Eqns.18, Eqns.19 respectively. Dynamic principles and control strategies set in place shall ensure stability and maintain the desired trajectories of flights.

The full transformation matrix \mathbf{T} combines rotation and translation, and is expressed as:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix} \quad (23)$$

where \mathbf{R} is the 3×3 rotation matrix and $\mathbf{p} = [x \ y \ z]^T$ is the position vector. The last row $[0 \ 0 \ 0 \ 1]$ is added to make \mathbf{T} a homogeneous transformation matrix. Thus, the full transformation matrix \mathbf{T} is:

$$\mathbf{T} = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi & x \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi & y \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

The positions of the quadrotor's four rotors in its local frame are defined as:

$$\mathbf{p}_1 = \begin{bmatrix} \frac{\text{size}}{2} \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} -\frac{\text{size}}{2} \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{p}_3 = \begin{bmatrix} 0 \\ \frac{\text{size}}{2} \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{p}_4 = \begin{bmatrix} 0 \\ -\frac{\text{size}}{2} \\ 0 \\ 1 \end{bmatrix} \quad (25)$$

where size denotes the distance between the center of the quadrotor and each rotor. To find out the positions of the rotors in the global frame, these local frame coordinates are transformed by utilizing the transformation matrix T . This transformation maps the rotor positions from the local coordinate system to the global coordinate system, considering the quadrotor's orientation and position.

In quadrotor trajectory planning, quintic polynomials are used to generate smooth transitions between waypoints by defining precise position, velocity, acceleration profiles and T_P trajectory duration. These polynomials $p(t)$ are described by the general form

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \quad (26)$$

where the coefficients a_0 through a_5 are determined by solving the linear equations based on the boundary conditions given for the initial and final positions, velocities, and accelerations. Along with the initial and final positions of waypoints given, we will define a parameter T_P for the trajectory duration. It influences the quadrotor motion, whereby a larger T_P results in smoother but more slowly joining paths, and a smaller T_P gives faster transitions that provide abrupt motion or instability. In our case of the simulation, a quintic polynomial approach for the quadrotor trajectory is calculated under the initial and final position and T_P . The coefficients a ensure that the path satisfies all of the constraints imposed upon it, such as that to provide smoothness from start and end. Once obtained, these coefficients are used to calculate the position, velocity, and acceleration profiles at any given time, enabling the quadrotor to follow a smooth path. The position, velocity, and acceleration profiles provide the necessary references for the control algorithms to adjust the quadrotor's thrust and attitude to follow the trajectory accurately. Additionally, a PID control system is employed to fine-tune the quadrotor's response and eliminate error accumulation while following

the given trajectories. In this paper, the waypoints that initially fed into the quintic polynomials are hyperchaotic trajectories generated by the algorithm discussed in the previous section. The sequence of hyperchaotic points throughout consecutive planes along the contour is iterated to generate the quadrotor trajectory, with a specific value of the time parameter T_P defining the trajectory duration between consecutive points in sequence.

5. Results and simulations

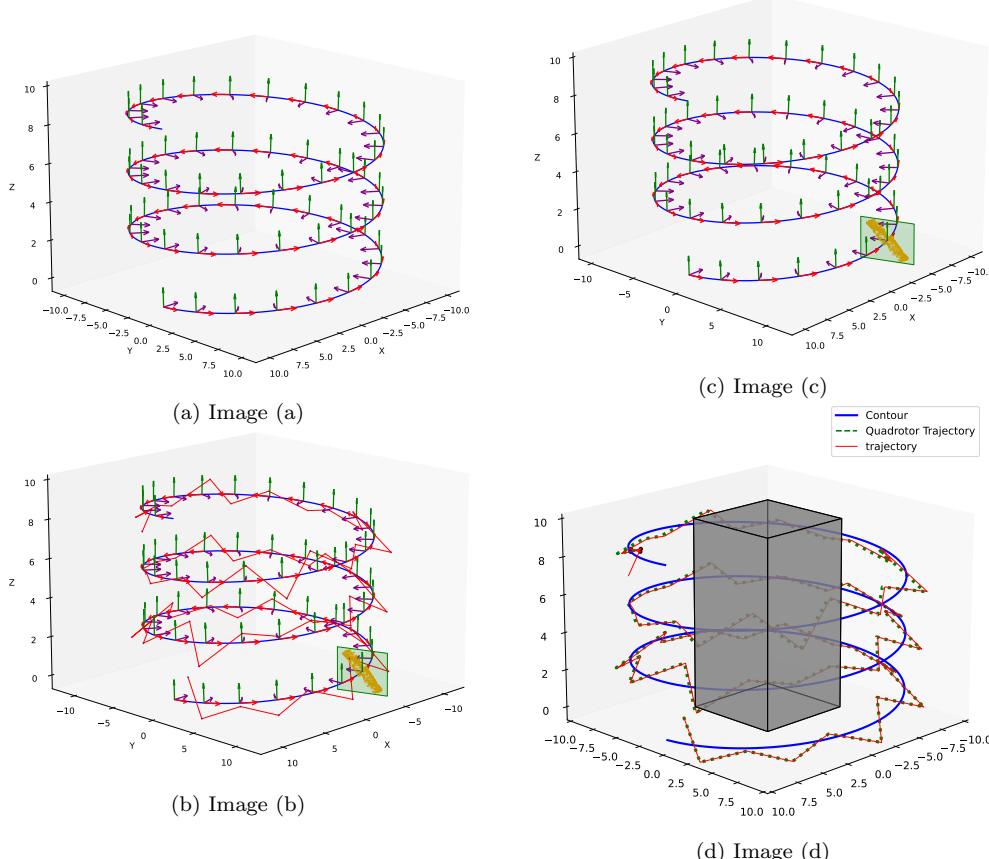


Figure 10: Common caption for all images.

In the initial phase of this study, a 3D open general helical contour was defined utilizing parametric equations Eq.11 with a pitch of 3 and three turns.

In order to avoid congestion, chosen the step parameter of 60, which makes such numbers of equidistant moving coordinates plotted along the helical contour, as illustrated in Fig.10a. On each plane within the moving frame, the 2D Q-DM hyperchaotic map was mapped using the reduced kinematic relative equation provided in Eq.15, which is shown only in the seventh plane for better observation in Fig.10b as given above. Note that labels for moving frames, contours, and planes are consistent throughout the paper, with the same colours and labels used uniformly. As a result, legends are not included in the figures. And started the iteration from 49000 in order to get a higher divergence of the sequence of the points and generated hyperchaotic trajectory is shown in Fig.10c as red lines. Fig.10d shows the Quadrotor trajectory, simulated corresponding to the given conditions with trajectory duration of $T_P = 40$ and mass of 200g. The Green dotted line shows the path traced by a quadrotor, and instead of building for simulation purposes, it shows a grey rectangular box, which is surrounded by a helical contour.

Scaling is a special case of affine transformation characterized by its linearity, where all the coordinates are multiplied by a scalar factor and with no translation involved, as given below

$$\mathbf{x}' = \mathbf{S}\mathbf{x} + \mathbf{b} = \mathbf{S}\mathbf{x}, \text{ where } \mathbf{b} = 0 \quad (27)$$

i.e The hyperchaotic equations of Q-DM map is termed as,

$$\begin{bmatrix} x_{n+1} \\ q_{n+1} \end{bmatrix} = \mathbf{S} \begin{bmatrix} k(q_n^2 - 1)x_n \\ q_n + x_n \end{bmatrix} \quad (28)$$

In that case, this paper suggests that after scaling, a hyperchaotic map still retains both its Lyapunov exponents positive because of the affine nature of this transformation. It is already shown by Strogatz et al. [39] that an affine transformation does not alter the hyperchaotic properties of a map, and thus, the intrinsic chaotic dynamics of it is preserved. Here, visual validation is provided through a closed circular contour of radius 3 and a step size of 25 for better clarity. In Fig.11a, the seventh plane is plotted along the contour and mapped the 2d NDMH map on it for observing scaling properties. In this Fig.11a, the sequence of hyperchaotic points is adapted by plotting the nth point on the nth plane to generate a trajectory shown as red lines. In Fig.11c, it illustrates the bifurcation plot of the 2D NDMH map determined with respect to the global z' coordinate and iterated for each value in the range of the control parameter u from 0 to 1. The global reference frame

is anchored at the origin of the circular contour, with no orientation of the contour relative to the z' global axis. Consequently, in the bifurcation plot, the trajectory variation in the z' direction is shown, which indicates proximity around the contour spans from -1.65 to 1.65 with no scaling applied or $S = 1$ in Eq.28.

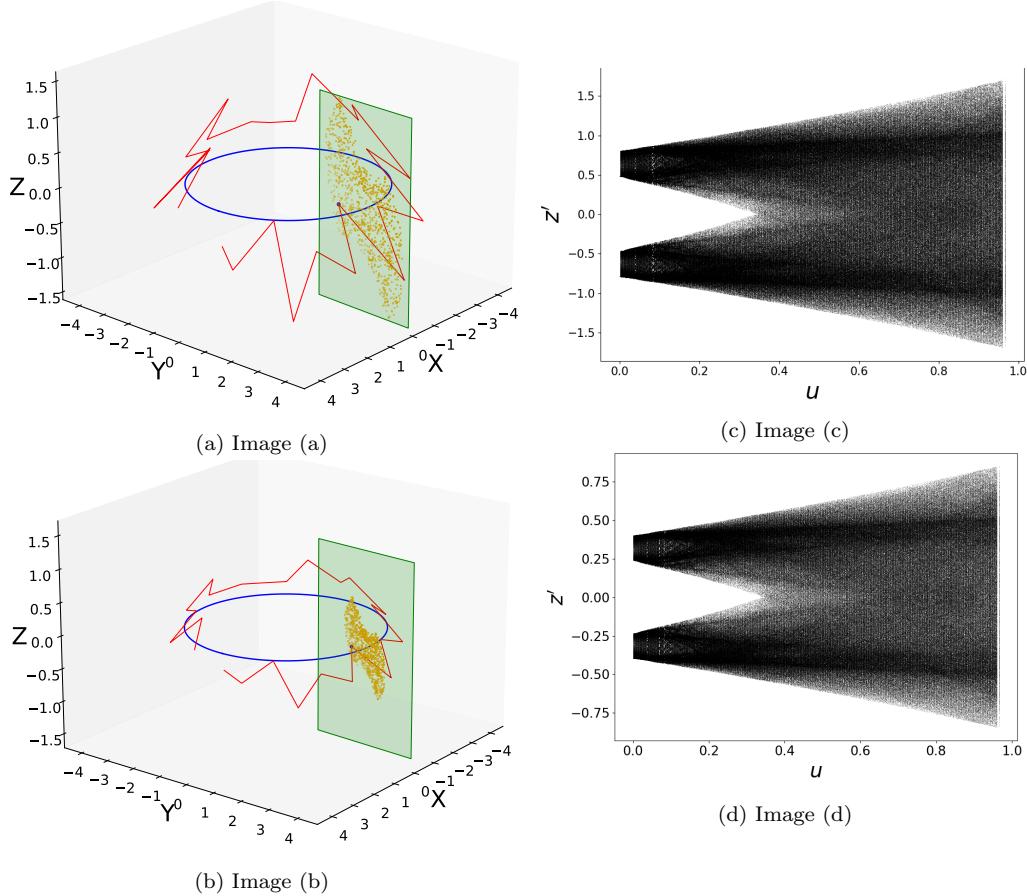


Figure 11: Common caption for all images.

In Fig.11b, the scaled trajectory with $S = 0.5$ is plotted along the contour and shows that the map in this plane is significantly less covered than for the unscaled version. Hence, the range of variation of z' in the bifurcation diagram reduced to -0.82 to 0.82 , in contrast to the earlier unscaled trajectory, which ranged from -1.65 to 1.65 . This then diminished the proximity dis-

tance from 3.30 to 1.65 units. this adaptation of scaling factor S ensures that the quadrotor would not leave the specified proximity of the contour, and also varying S can adjust the proximity for necessities like obstacle avoidance.

The objective of this paper is to derive multiple quadrotor hyperchaotic trajectories that are mutually independent, which enables multiple quadrotors for boundary surveillance around vertical structures. The main purpose of these hyperchaotic trajectories is to provide the ability to operate more quadrotors simultaneously due to their independence and divergence. Surveillance around a stadium-like structure with two turns of helical contour progressing vertically along the Z' axis was considered and is shown in Fig.12a. In Fig.12b, the moving coordinates are plotted with simulation parameter steps, which is set to 40, and this value was chosen for better visual observation. A single plane is also plotted along the contour to illustrate that the trajectory is part of a 2D NDMH map. In Fig.12c, two hyperchaotic trajectories with slightly different initial conditions (0.1,0.1) and (0.11,0.11) are plotted. Trajectory 1 and trajectory 2 are represented by the red and orange lines, respectively. Both hyperchaotic trajectories were started after 49,000 iterations to discard transient behaviour and achieve higher divergence. Further trajectories included may overcrowd Fig.12c, hence making it difficult to understand.

In Fig.12d, the green dotted line shows the trajectory traced by the simulated quadrotor model. An efficient control system has been utilized to minimize the noise as well as accumulated errors for precise adherence to the hyperchaotic trajectory. The quadrotor tracks the desired trajectory without significant deviations, due to the fact that the control parameter T_P is set to a trajectory duration of 60. Larger values of T_P provide greater stability and slower joining paths. Therefore, a slower profile was chosen for quadrotor tracing. Also, Fig.12e shows the quadrotor following the second trajectory, with the magenta dotted line indicating its path traced.

6. Real-world implications

Sudden changes in the quadrotor's trajectory at waypoints can result in high variation of linear and angular acceleration in very short periods, which introduces extra noise in acceleration and gyroscope sensors[43]. This is because accelerometers are naturally very susceptible to mechanical oscillations due to internal moving parts. It is possible to mitigate this issue by maintaining a stable distance between planes, ensuring smooth and stable quadrotor

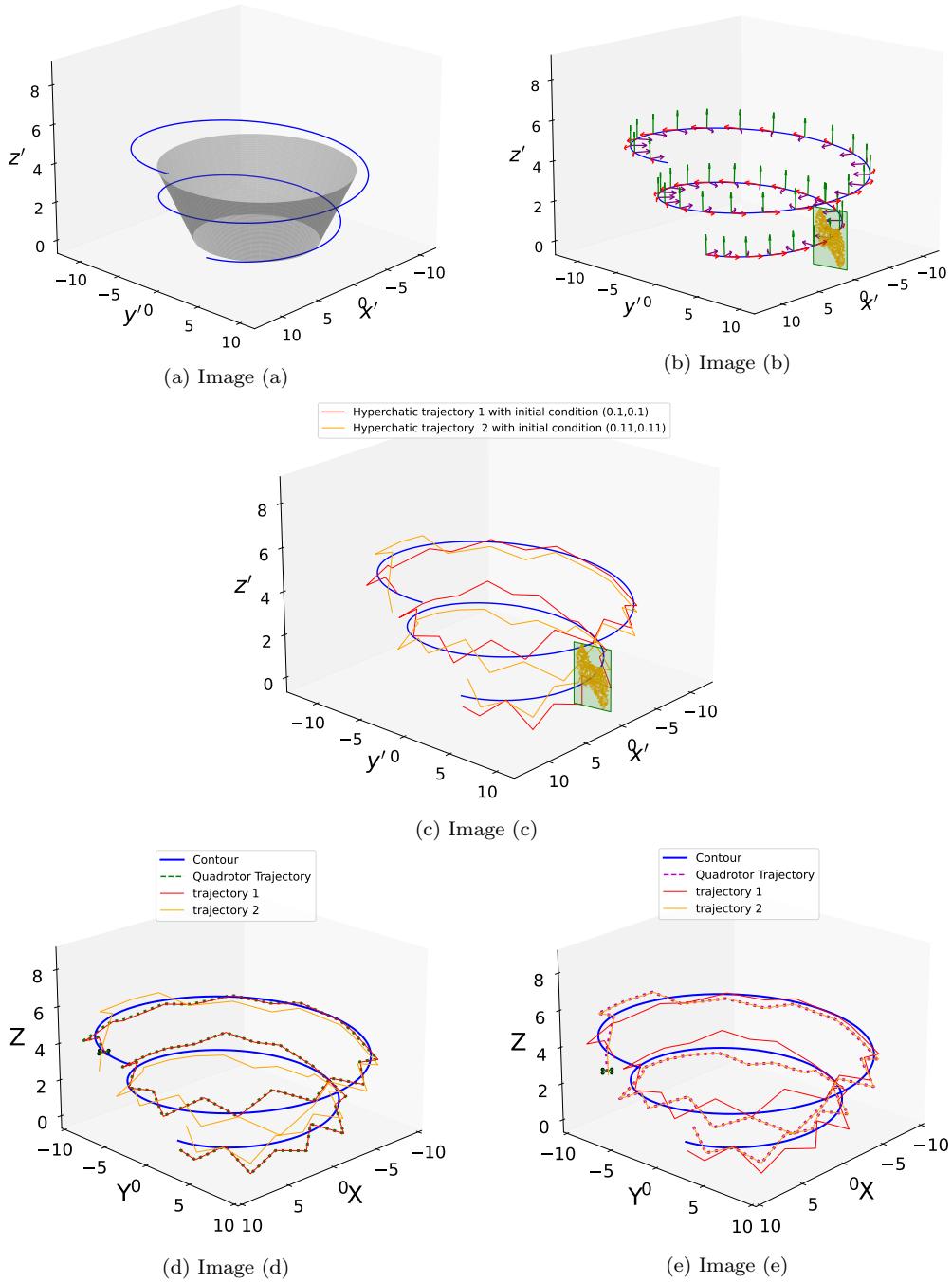


Figure 12: Main caption describing all the images.

motion. This will, in turn, help in efficient tracing of the hyperchaotic trajectory. And some other methods are also available , which may be applied for noise reduction.

7. Conclusion

This paper presents the hyperchaotic path planning for a quadrotor to do boundary surveillance missions under adversarial conditions. From 2D contours, upgrading to helical 3D contours can substantially improve the effectiveness and unpredictability of boundary surveillance and enhance coverage while flying adaptively around vertical structures like buildings. Based on our algorithm, the rotation and sliding of 2D hyperchaotic maps create 3D trajectories along the helical contour. The proposed method provides a significant advance from previous approaches of using chaotic maps in 2D robot trajectory planning to 3D applications by introducing a hardware-realizable hyperchaotic NDMH map. And parametrically evaluated the high divergence of the NDMH system with others. In real-world applications, considering obstacle avoidance and constraints around boundaries are crucial. Since the hyperchaotic system supports scaling affine transformation, it can also serve that purpose by trajectory adaptation. The results of this article show that the generated hyperchaotic trajectories are mutually independent. So a quadrotor can trace these hyperchaotic trajectories with high precision if an effective control system is employed. More importantly, the methodology proposed to generate numerous hyperchaotic trajectories, which allows multiple quadrotors to trace them simultaneously. This results in enhanced unpredictability, particularly valuable for sensitive surveillance scenarios. Incorporating real-time obstacle avoidance mechanisms within the quadrotors will ensure simultaneous multi-quadrotor surveillance without any interference between them or with their surroundings. This further increases the reliability of their operation.

References

- [1] D. Huamanchahua, D. Yalli-Villa, A. Bello-Merlo, J. Macuri-Vasquez, Ground robots for inspection and monitoring: A state-of-the-art review, in: 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE, 2021, pp. 0768–0774.

- [2] W. Budiharto, Intelligent surveillance robot with obstacle avoidance capabilities using neural network, *Computational intelligence and neuroscience* 2015 (1) (2015) 745823.
- [3] A. Nayyar, V. Puri, N. G. Nguyen, D. N. Le, Smart surveillance robot for real-time monitoring and control system in environment and industrial applications, in: *Information Systems Design and Intelligent Applications: Proceedings of Fourth International Conference INDIA 2017*, Springer, 2018, pp. 229–243.
- [4] D. S. Terracciano, L. Bazzarello, A. Caiti, R. Costanzi, V. Manzari, Marine robots for underwater surveillance, *Current Robotics Reports* 1 (2020) 159–167.
- [5] H. M. Do, M. Pham, W. Sheng, D. Yang, M. Liu, Rish: A robot-integrated smart home for elderly care, *Robotics and Autonomous Systems* 101 (2018) 74–92.
- [6] Z. Zhou, C. Zhang, C. Xu, F. Xiong, Y. Zhang, T. Umer, Energy-efficient industrial internet of uavs for power line inspection in smart grid, *IEEE Transactions on Industrial Informatics* 14 (6) (2018) 2705–2714.
- [7] M. Benedict, J. Mullins, V. Hrishikeshavan, I. Chopra, Development of a quad cycloidal-rotor unmanned aerial vehicle, *Journal of the American Helicopter Society* 61 (2) (2016) 1–12.
- [8] M. Idrissi, M. Salami, F. Annaz, A review of quadrotor unmanned aerial vehicles: applications, architectural design and control algorithms, *Journal of Intelligent & Robotic Systems* 104 (2) (2022) 22.
- [9] G. Wang, X. Wang, S. Li, K. Lu, Order-supplementary finite-time trajectory tracking control of quadrotor unmanned aerial vehicles, *Nonlinear Dynamics* 112 (10) (2024) 8229–8247.
- [10] P. Chang, S. Yang, J. Tong, F. Zhang, A new adaptive control design for a quadrotor system with suspended load by an elastic rope, *Nonlinear Dynamics* 111 (20) (2023) 19073–19092.
- [11] B. Zhou, F. Gao, L. Wang, C. Liu, S. Shen, Robust and efficient quadrotor trajectory generation for fast autonomous flight, *IEEE Robotics and Automation Letters* 4 (4) (2019) 3529–3536.

- [12] N. Imanberdiyev, C. Fu, E. Kayacan, I.-M. Chen, Autonomous navigation of uav by using real-time model-based reinforcement learning, in: 2016 14th international conference on control, automation, robotics and vision (ICARCV), IEEE, 2016, pp. 1–6.
- [13] W. Zhao, H. Liu, F. L. Lewis, K. P. Valavanis, X. Wang, Robust visual servoing control for ground target tracking of quadrotors, *IEEE Transactions on Control Systems Technology* 28 (5) (2019) 1980–1987.
- [14] W. Kwon, J. H. Park, M. Lee, J. Her, S.-H. Kim, J.-W. Seo, Robust autonomous navigation of unmanned aerial vehicles (uavs) for warehouses' inventory application, *IEEE Robotics and Automation Letters* 5 (1) (2019) 243–249.
- [15] C. Wang, J. Wang, Y. Shen, X. Zhang, Autonomous navigation of uavs in large-scale complex environments: A deep reinforcement learning approach, *IEEE Transactions on Vehicular Technology* 68 (3) (2019) 2124–2136.
- [16] J. Zheng, R. Chen, T. Yang, X. Liu, H. Liu, T. Su, L. Wan, An efficient strategy for accurate detection and localization of uav swarms, *IEEE Internet of Things Journal* 8 (20) (2021) 15372–15381.
- [17] D. Saldana, R. J. Alitappeh, L. C. Pimenta, R. Assunção, M. F. Campos, Dynamic perimeter surveillance with a team of robots, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2016, pp. 5289–5294.
- [18] S. N. Elaydi, *Discrete chaos: with applications in science and engineering*, Chapman and Hall/CRC, 2007.
- [19] A. Akgul, I. Moroz, I. Pehlivan, S. Vaidyanathan, A new four-scroll chaotic attractor and its engineering applications, *Optik* 127 (13) (2016) 5491–5499.
- [20] A. Akgul, H. Calgan, I. Koyuncu, I. Pehlivan, A. Istanbullu, Chaos-based engineering applications with a 3d chaotic system without equilibrium points, *Nonlinear dynamics* 84 (2016) 481–495.

- [21] X. Zang, S. Iqbal, Y. Zhu, X. Liu, J. Zhao, Applications of chaotic dynamics in robotics, *International Journal of Advanced Robotic Systems* 13 (2) (2016) 60.
- [22] C. Nwachioma, J. H. Pérez-Cruz, Analysis of a new chaotic system, electronic realization and use in navigation of differential drive mobile robot, *Chaos, Solitons & Fractals* 144 (2021) 110684.
- [23] S. Nasr, H. Mekki, K. Bouallegue, A multi-scroll chaotic system for a higher coverage path planning of a mobile robot using flatness controller, *Chaos, Solitons & Fractals* 118 (2019) 366–375.
- [24] S. Liu, Y. Wei, J. Liu, S. Chen, G. Zhang, Multi-scroll chaotic system model and its cryptographic application, *International Journal of Bifurcation and Chaos* 30 (13) (2020) 2050186.
- [25] P. S. Gohari, H. Mohammadi, S. Taghvaei, Using chaotic maps for 3d boundary surveillance by quadrotor robot, *Applied Soft Computing* 76 (2019) 68–77.
- [26] F. A. Hossam Eldein Mohamed, W. El-Shafai, Cancelable biometric authentication system based on hyperchaotic technique and fibonacci q-matrix, *Multimedia Tools and Applications* (2024) 1–39.
- [27] H.-w. Xue, J. Du, S.-l. Li, W.-j. Ma, Region of interest encryption for color images based on a hyperchaotic system with three positive lyapunov exponents, *Optics & Laser Technology* 106 (2018) 506–516.