IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 October 2023, accepted 20 November 2023, date of publication 28 November 2023, date of current version 5 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337371

SURVEY

Chaotic Motion Planning for Mobile Robots: Progress, Challenges, and Opportunities

FARIDA AHURAKA[1], PATRICK MCNAMEE[2], QIXU WANG[2], ZAHRA NILI AHMADABADI[2], (Member, IEEE), AND JEFFREY HUDACK[3], (Member, IEEE)
[1]Department of Mechanical Engineering, Wichita State University, Wichita, KS 67260, USA
[2]Department of Mechanical Engineering, San Diego State University, San Diego, CA 92182, USA
[3]Information Directorate, Air Force Research Laboratory, Rome, NY 13441, USA

Corresponding author: Zahra Nili Ahmadabadi (zniliahmadabadi@sdsu.edu)

This work was supported in part by Wichita State University under Grant U15096, and in part by the Air Force through memorandum of understanding (MOU) under Grant FA8750-15-3-6000. The U.S. Government is authorized to reproduce and distribute copies for Governmental purposes notwithstanding any copyright or other restrictive legends. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force or the U.S. Government.

ABSTRACT Chaotic path planners are a subset of path planning algorithms that use chaotic dynamical systems to generate trajectories throughout an environment. These path planners are imperative in surveillance tasks in the presence of adversarial agents which require the paths to be unpredictable while at the same time guaranteeing complete coverage of the environments. In the online coverage of unknown terrain, the chaotic path planning algorithms can work without the need of the environment map and the designer has additional control over the generated paths relative to other heuristic coverage path planners such as random-walk algorithms. Although chaotic path planners have been studied over the past two decades, there has not been an updated survey on the advances. This paper presents an up-to-date review by providing: an introduction of commonly used chaotic systems and methods for their manipulation; an overview of obstacle avoidance methods used by chaotic path planners; and a discussion on other applications, challenges, and research gaps.

INDEX TERMS Autonomous robots, chaos, path planning, nonlinear dynamical systems, robot motion.

I. INTRODUCTION

Chaos is all around us. From weather [1], [2], [3] to population growth [4], [5], the unpredictable nature of chaos prevents predicting any phenomenon associated with it. Chaos can be defined as highly unpredictable behavior induced by extreme sensitive dependence on initial conditions [6]. In certain nonlinear systems, this sensitive dependence creates a set of rich dynamical behaviors that are critical to numerous fields, including Biomedical Engineering [7], Nursing [8], and Psychology [9]. Robotic engineering is also one field that has been influenced by chaos theory. The application of chaos theory to robotic path planning has been studied over the past two decades.

Path planning is the task of constructing a collision free path between two points in a workspace [10], [11]. For the purpose of this paper, the path planning methods are broadly classified into two categories: point-to-point path planning [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42] and coverage path planning [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61]. The point-to-point path planning problem involves finding a collision-free trajectory from some starting position to a goal or final position. Sampling-based planning [62], [63], [64], [65], [66], [67], [68], kinodynamic planning [69], [70], [71], [72], [73], and manipulation planning [74], [75], [76], [77], [78] are among the methods used in studies to find a direct path that connects two points in space.

Coverage path planning [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], as implied by the name, involves finding a path that passes through all points in an area. The robots must completely cover the area while avoiding any obstacles and minimizing overlapping or repeating paths in an optimal manner [90]. The coverage problem is an

The associate editor coordinating the review of this manuscript and approving it for publication was Ludovico Minati.

NP-hard problem as it encompasses finding a collision-free path while simultaneously ensuring that every space has been covered, preferably only once. In terms of adaptability to unknown environments, CPP algorithms are classified into offline algorithms and online algorithms. Offline algorithms [91], [92], [93], [94], [95], [96], [97], [98], [99], [100], [101], [102], [103], [104], [105], [106] need full knowledge of the map and any obstacles in the space must be stationary. The assumption of *a priori* known environments will usually lead to more efficient coverage, but this assumption is not applicable to many dynamic real-world situations. Conversely, online algorithms [107], [108], [109], [110], [111], [112], [113], [114], [115], [116], [117], [118], [119], [120] do not require full knowledge of the map in advance; they make use of advancements in collision detectors and real-time measurements to ensure full coverage of the reachable target space [90]. Online CPP algorithms are well suited to automated mobile robots and their numerous applications. In terms of algorithm complexity and sensory needs, the online CPP algorithms can be divided into two classes: complete and heuristic [80]. According to [80], the complete CPP algorithms mostly use cellular decomposition methods, either explicitly or implicitly, to generate deterministic simple paths that cover the target environment to achieve complete coverage. Both explicit and implicit complete CPP algorithms require a robot with localization capabilities. Alternatively, heuristic CPP algorithms, such as random CPP or chaotic CPP, can generate unpredictable paths to cover completely unknown spaces without any localization capabilities. Compared to the cellular decomposition methods, the random and chaotic CPP use algorithms with less explicit programming, leading to a reduction in required computational power and complexity. However, both random and chaotic algorithms cannot guarantee complete coverage if they are not tuned properly to the operational environment(s). This issue can be fixed in case of chaotic planners. Unlike random planners, chaotic planners generate unpredictable but yet deterministic paths. This determinism allows the system designer to control the chaotic paths and develop approaches to guarantee the full coverage of an unknown space.

The first paper on chaotic path planning was published in 2001 by Nakamura and Sekiguchi [121]. Subsequently, chaotic path planners became a subject of interest because of their rich array of applications ranging from domestic uses, e.g. vacuum robots, to military applications, e.g. autonomous surveillance robots. For a path planning algorithm to be classified as a chaotic path planning algorithm, it must rely on a chaotic dynamical system (CDS) or maps for generating robotic trajectories in the environment. A dynamical system is chaotic if it displays sensitive dependence to initial conditions [122], [123] and topological transitivity. Sensitive dependence implies that a minuscule change in the initial conditions of the dynamical system lead to drastic changes in the trajectories of the dynamical system. The topological transitivity characteristic implies that the domain of the dynamical system cannot be subdivided into isolated regions. Any trajectory in the dynamical system that starts in one region of the domain will always end up in a different region of the domain. Chaotic path planners will in some way use dynamical systems with these two characteristics to generate trajectories in the robot's operating environment e.g., projecting a three dimensional chaotic dynamical system into a two dimensional room that a robot is exploring. While the topological transitivity property can be used to guarantee complete coverage of the entire environment over time by some chaotic CPP, it does not normally ensure the uniformity and efficiency of the robot's coverage. In the absence of control over chaotic paths, the generated paths might lead to repetitive coverage that would significantly increase the coverage time. This paper will explore various manipulation methods aimed at improving the uniformity and efficiency of chaotic paths.

Chaotic path planning has the potential to reinvigorate the field of robotics for online coverage of unknown environments. However, only a limited number of studies explore aspects of chaotic path planners for this task such as [124] which reviewed various applications of chaos in robotics and only briefly discusses chaotic path planning. This current study aims to conduct a thorough survey of the previous works specifically done on chaotic path planners to provide a starting point for future investigations within this area. The following sections of this work are structured to provide brief overview of non-chaotic coverage algorithms for context on the coverage task before introducing chaotic dynamical systems with manipulations and applications found in previous works. Section II is the overview of non-chaotic coverage path planning methods proceeding section III which explores the various chaotic path planning algorithms that have been used in previous work. Section IV explores various chaos manipulation strategies which seek to improve the uniformity and efficiency of coverage while section V describes other applications of chaotic dynamical systems for path planning apart from CPP. In section VI, the key topic of obstacle avoidance will be introduced and analyzed. Lastly section VII explores current problems facing chaotic path planning applications before recommendations for future studies and a brief conclusion given in section VIII.

II. OVERVIEW OF NON-CHAOTIC PLANNERS

This section gives a brief overview of popular non-chaotic CPP algorithms as an introduction to the coverage task and its intricacies. Both complete and heuristic CPP are reviewed in the following section.

A. COMPLETE CPP – CELLULAR DECOMPOSITION METHODS

Complete CPP algorithms are coordinated methodical efforts to cover an area. They mostly use cellular decomposition methods that decompose the free space into non-overlapping cells. These cells are considered simple to cover since they do not contain any obstacles. The complete coverage is achieved once the robot visits all the cells. The previous works [80], [90] present a comprehensive review of the studies on cellular decomposition methods. This section is not intended to be an exhaustive overview of the advances made in this area; it is rather a brief review of some of the important methods to enable comparison with chaotic algorithms. Cellular decomposition approaches can be classified into three categories [80]: 1) approximate decomposition; 2) semi-approximate decomposition; and 3) exact decomposition.

1) APPROXIMATE CELLULAR DECOMPOSITION

The approximate methods break down the target space into fine and uniform grid cells such that the collection of these cells approximates the shape of the target space. The cells are the same size of the robot's physical footprint so that when a robot enters a cell, the cell can be considered visited. The most popular algorithms proposed under this category include: 1) the wavefront algorithm [94], [125]; 2) spanning tree covering (STC) [107], [110], [126], [127]; and 3) neural network [111], [112], [128], [129], [130], [131]. In all these methods, the algorithm is recursively called to select an unvisited neighboring cell for the robot to visit next. Their primary difference lies in the methods used to prioritize and rank the unvisited neighboring cells. The prioritization attempts to improve the energy and time efficiency by reducing the path length and/or the number of turns. For instance, the wavefront algorithm propagates a distance wave front, through the free space, from an arbitrary point in the space called the "goal". The wave front travels around the obstacles and assigns a number to each cell, proportional to their distance from the goal. The robot prioritizes visiting the neighboring cells which lie the furthest away from the goal as shown in Fig. 1a.

Alternatively, STC selects the next neighboring cell by moving along both sides of a spanning tree. Two different cell sizes are used, a large and a small cell size although the large cells are sometimes referred to as mega cells. Small cells are obtained by dividing the large cells into four parts, with each small cell being the size of the robot's footprint. The spanning tree is constructed incrementally using the onboard sensors and by connecting the centers of adjacent free mega cells. Using this algorithm, the robot never visits any of the small cells twice except for the starting cell as can be seen in Fig. 1b. Lastly, the neural network algorithm selects the next cell to visit based on comparing the "neural" activity of the current cell i.e., neuron, and that of the neighbors. The neighbors with the largest neural activity are selected as the next cell to be visited. In the context of a cleaning robot, the unclean cells i.e., unvisited cells, have the highest neural activity while obstacles have the lowest. This high neural activity will attract the robot, leading to the complete coverage of the space.

2) SEMI-APPROXIMATE CELLULAR DECOMPOSITION

The semi-approximate cellular decomposition algorithms differ from approximate cellular decomposition algorithms in that the semi-approximate algorithms do not need *a prior* information about the environment [113], [132], [133]. These path planners will decompose the space into vertical slices of the identical width with the top and bottom boundaries of these slices, or cells, having any shape. The robot starts from an arbitrary point in the space and zigzags along the grid lines created by the cells to cover the area. This zigzagging might miss covering some smaller areas, called inlets, or cover them twice. The algorithm would detect those inlets as well as inlets within inlets and cover them through by calling the same zigzag procedure recursively. Fig. 2 shows the robot's path when it uses the semi-approximate cellular approximation algorithm suggested by [113].

3) EXACT CELLULAR DECOMPOSITION

Whereas approximate and semi-approximate cellular decomposition methods attempted to use a grid structure or regular pattern to partition an environment into cells, exact cellular decomposition uses the obstacles in the environment to form cells. The exact cellular decomposition methods divide the free space into connected cells and store the decomposition as an adjacency graph with two cells considered adjacent if they share a common boundary. The cells are generated by sweeping a slice through the space with cell boundaries formed when the sweeping slice encounters some event, i.e. a change in its connectivity due to emergence or disappearance of an obstacle. After all the sweeps are complete, the path planning algorithm will compute a sequence of cells to be visited through the adjacency graph. Following this sequence will guarantee the complete coverage as the robot will visit each cell exactly once. Once the robot enters a cell, the algorithm will generate a specific zigzag path for it to cover that cell.

Over the years, researchers have proposed three major versions of exact algorithms: 1) trapezoidal decomposition; 2) Boustrophedon decomposition; and 3) Morse-based decomposition. The difference between these algorithms lies in the configurations of the cells as well as the type of the environments (and obstacles) that each algorithm can handle. As the name would suggests, the trapezoidal decomposition algorithm creates only trapezoid-shaped cells and is well suited to only handle planar, polygonal, and *a priori* known spaces. As shown in Fig. 3a, the trapezoidal decomposition can only create convex cells which can result in some inefficiencies. Many of the cells formed by the trapezoidal algorithm can be merged to have more efficient coverage paths. The Boustrophedon algorithm [134], [135] was developed to address this issue by allowing creation of non-convex cells which reduces the number of cells as observed in Fig. 3b. Similar to trapezoidal decomposition, Boustrophedon decomposition is classified as an offline method and can only handle polygonal obstacles within the environment.

To address these limitations, [136] generalized the Boustrophedon decomposition using the critical points of Morse functions [137] and developed the Morse-based

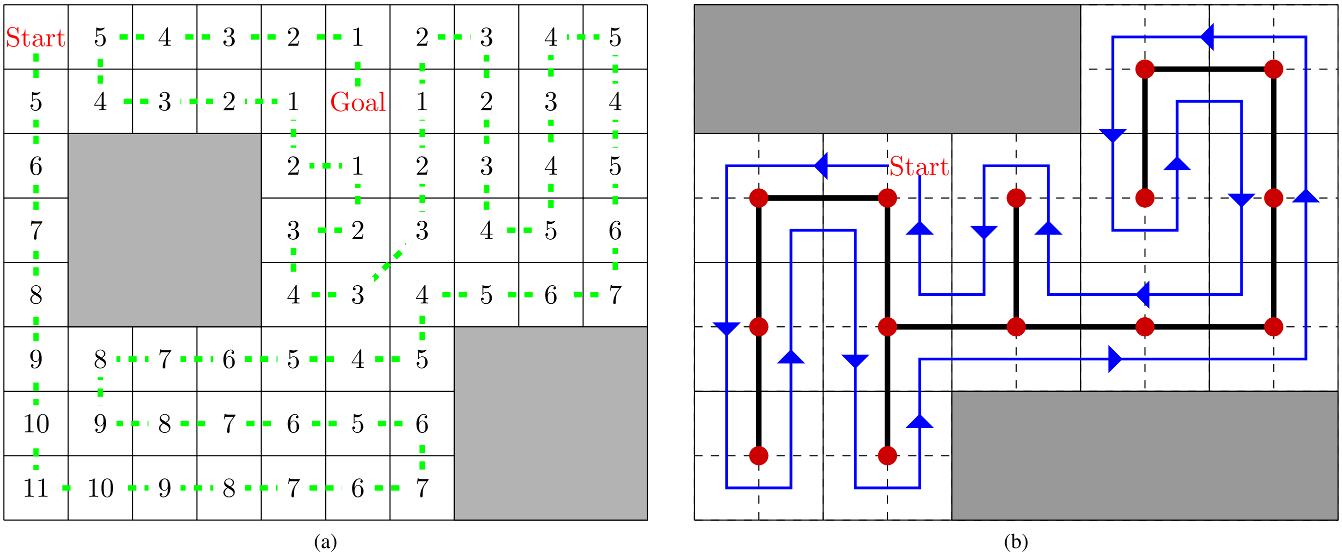


FIGURE 1. **Coverage of a space using approximate cellular decomposition with obstacles shown as filled rectangles. (a) Wavefront algorithm with path shown as dashed line. (b) STC with path shown as solid blue line.**

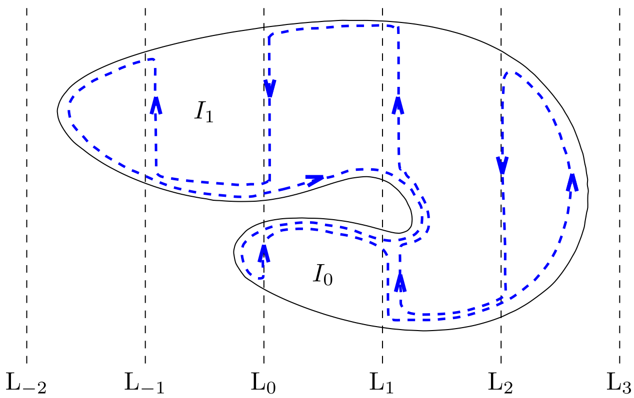


FIGURE 2. **Robot's path generated using the semi-approximate cellular decomposition. The robot identifies and covers inlets I_0 and I_1.**

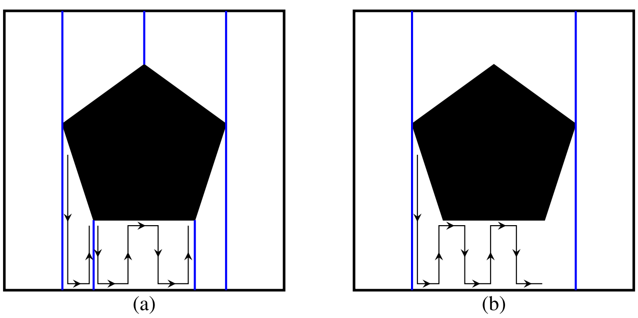


FIGURE 3. **Robot paths generated using (a) trapezoidal and (b) Boustrophedon algorithms.**

decomposition. The critical points are located on the boundary of an obstacle where the surface normal vector of the obstacle is perpendicular to the sweep line. The connectivity of the sweep line will change when it meets a critical point as shown in Fig. 4a and 4b. As the sweep line intersects the obstacle at the first critical point, its connectivity changes from one to two and therefore two new cells are formed. The appearance of the second critical point indicates the emergence of an obstacle-free region and therefore the two cells are closed, and one new cell is created. In spaces unknown at the start of the coverage task, the robot can use its sensors to detect the critical points of the obstacles [138], [139] and efficiently determine the cell boundaries. However, the sensors can only detect these points when their distance from the robot is minimal. This takes place only when the sweeping direction is parallel to the normal to the obstacle surface, i.e., the robot is following the obstacle wall. Therefore, to ensure detection of the critical points, the robot should start following the obstacle walls as soon as it detects an obstacle, and it should continue doing so until either the sensors detect a critical point, or it reaches the next strip [136] as shown in Fig. 4c.

B. HEURISTIC CPP – RANDOM

The CPP algorithms discussed in Section II-A use deterministic algorithms which have the drawback of requiring either an offline path generation or a large amount of memory to store long paths in vast environments. In contrast, the robot can heuristically cover an area by simply selecting random directions to travel within the search space at specified steps or distances. This concept forms the basis of random search methods (also known as random-walk), that do not require the memory resources to store previous trajectory of the robot. Random-walk (RW) methods can either be uncorrelated walks where the direction of travel chosen independent of the previous directions or correlated walks with the walk direction biased towards a preferred direction or a given target [140]. In robotics applications, uncorrelated RW methods

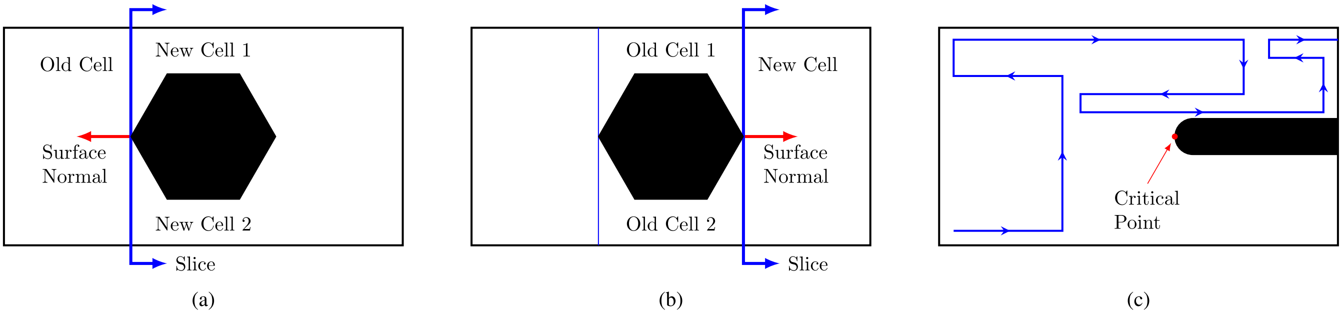


FIGURE 4. (a,b) Cellular decomposition using the Morse algorithm and (c) rectangular cycles within path enable detection of critical points.

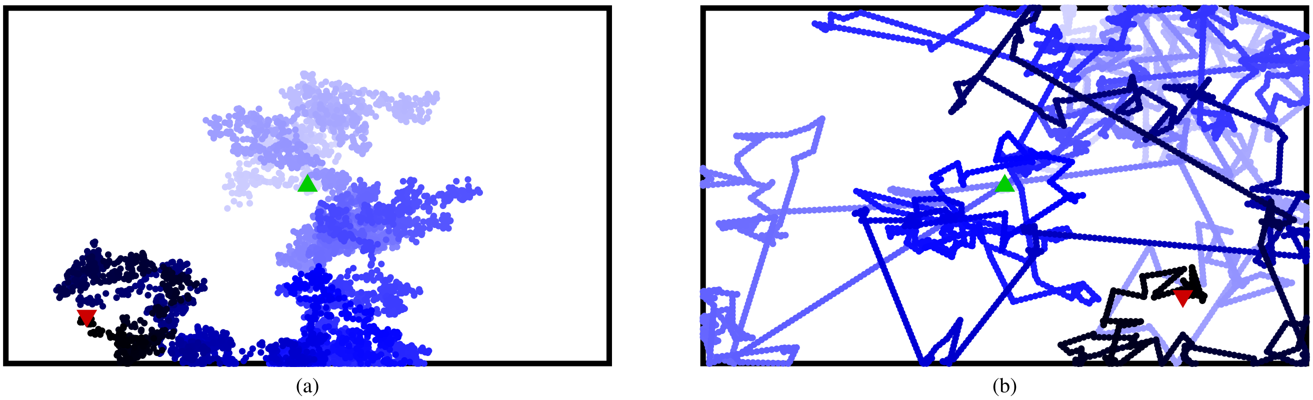


FIGURE 5. Paths generated using (a) Brownian motion and (b) Lévy flight to explore an enclosed space over a finite time interval. The paths have the same starting point ▲ but different ending points ▼.

such as Brownian motion (BM) and Lévy flight (LF) are commonly used [141], [142]. Comparing the two methods, the BM method takes small steps which are more efficient for local search as shown in Fig. 5a whereas the LF method is more successful in global search as it allows the agent to take occasional long steps and visit remote locations as shown in Fig. 5b. In ecological examples, animals foraging for food in food sparse environments resemble LF methods [143] but when the food is abundant, the simple BM method proves to be sufficient [144], [145].

Unlike the cellular decomposition methods, random search strategies do not guarantee complete coverage although random searches do not need localization sensors or as high of a computational capability. The cost of providing sophisticated navigation capabilities for agents with coordinated search can be prohibitive particularly for robotic swarms in remote and/or adversarial environments where there is a higher possibility of losing robotic agents. References [145] and [146] argue that if the cost of building agents with advanced navigation capabilities is higher than the price paid for the reduced performance, then it may be beneficial to use the random search strategies.

One major shortcoming of the random search is the lack of continuity in the robot's motion; the generated paths often result in sharp turns and uneven density of coverage across the space. While utilizing teams of robots can help alleviating the coverage density issue, strategies should be used to distribute the agents evenly across the space and improve search efficiency of the RW methods. Studies have used variety of approaches to realize this capability such as: 1) use of pheromones for the agents to communicate with each other [147], [148]; 2) combining the RW with an artificial potential field to disperse neighbouring robots [149]; and 3) adaptively switching between LF and BM [150] to cover the area both locally and globally. Regardless of the methods used to improve the coverage density, the discontinuous nature of RW when determining directions to travel means that sharp angular turns cannot be avoided. The chaotic path planners can present a simple solution to this issue. Continuous chaotic systems have the capability to generate trajectories with smooth turns and more uniform coverage compared to RW methods. In addition, chaotic search can also be implemented on low-cost robots in case of limited sensing and computational resources in much the same way as random CPP.

III. CHAOTIC SYSTEMS

Since the first applications of chaotic dynamical systems in path planning tasks, several dynamical systems with chaotic behaviors have been explored. Some of these systems, as will be explored in this section, display chaotic properties exclusively under certain subset of parameters.

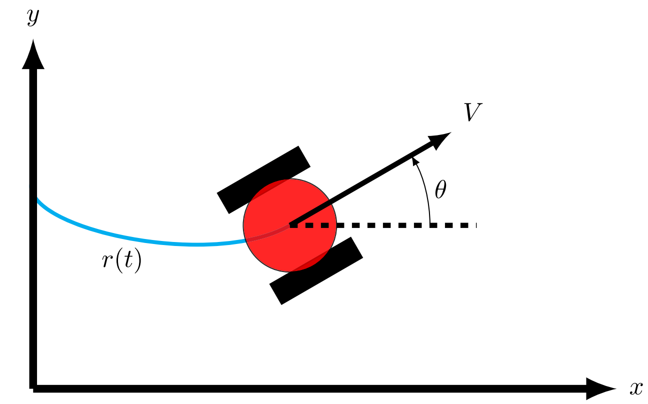


FIGURE 6. The standard two-wheeled mobile robot traveling in a plane along a path $r(t)$ with a vehicle speed V and velocity inclination θ.

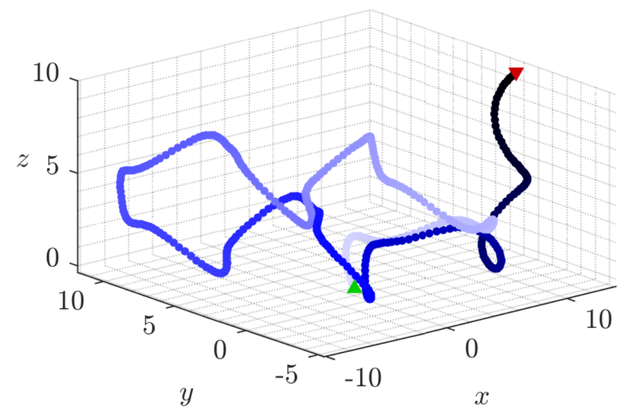


FIGURE 7. Chaotic Arnold 3D flow using the parameters $A = 1$, $B = 0.5$, and $C = 0.5$.

These systems are tested for chaos using bifurcation diagrams or the system's maximal Lyapunov exponent (MLE). The bifurcation diagram presents the qualitative changes in the system dynamics as parameters vary [151] while the MLE measures the sensitivity of the dynamical system to initial conditions. A dynamical system is considered chaotic if it falls within the chaotic regions of the parameter space in the bifurcation diagrams or if the MLE of the system is positive [6]. Once a dynamical system with appropriate parameters is selected, this chaotic dynamical system can be used to generate chaos in either the robot's controller behavior or in the robot's trajectory within the environment. The standard mobile robot used for simulation in most previous studies is a two wheel differential drive mobile robot. While this platform selection often restricts the environment to two dimensions, there have been studies involving application of chaotic dynamical systems in three dimensional environments [152]. Fig. 6 illustrates such a two wheeled differential drive robot comprised of two active fixed wheels and one passive caster wheel. The kinematics of the two-wheeled robot are subject to a non-holonomic constraint that prevent transverse motion. To directly induce chaos into the selected robot's dynamics, one or more coordinates of the dynamical system are often used to perturb or control the robot's velocity orientation. The result is that the robot steers in an unpredictable motion, traversing the environment.

The chaotic systems most commonly studied for use in CPP found within the literature include: 1) the Arnold system [153]; 2) the Lorenz system [154]; 3) the Chen and Ueta system [122], [155], and the Chen and Lü System [156]; 4) Chua's electrical circuit [157], [158]; and 5) the Logistic map [6]. A recent study [159] proposed as well a new chaotic system, though the paper was more focused on describing dynamic properties of this system rather than CPP performance. In the following sections, we will be describing the aforementioned commonly used chaotic systems by giving the equations governing the dynamical systems, some parameters that produce chaotic behaviors, and finally, a comparison between some of the systems. Throughout the section, there will be various plots of attractors where the system is simulated by a numerical method from an initial condition ▲ to generate points of the trajectory ● with the final point ▼ marked on the plot.

A. ARNOLD SYSTEM

One of the most researched and tested chaotic equations, the Arnold system is a nonlinear continuous dynamical system that was first used by [121] to realize chaotic path planning. Consequently, [121] developed a method of imparting desired chaotic motion into a mobile robot's controller by directly using one of the Arnold system state variables as the heading angle turning rate. The Arnold system is defined with parameters A, B, and C which are constants in (1).

$$\begin{cases} \dot{x} = A \sin z + C \cos y \\ \dot{y} = B \sin x + A \cos z \\ \dot{z} = C \sin y + B \cos x \end{cases} \tag{1}$$

The dot in (1), as well as the following equations, denotes differentiation of the states x, y, and z with respect to time. The Arnold equation describes a steady solution to 3D Euler Eqs. (2) and (3) which express the behavior of noncompressive perfect fluids on a 3D torus space.

$$\frac{\partial v_i}{\partial t} + \sum_{k=1}^{3} v_k \frac{\partial v_i}{\partial x_k} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + f_i \tag{2}$$

$$\sum_{i=1}^{3} \frac{\partial v_i}{\partial x_i} = 0 \tag{3}$$

where x_i and v_i denote the position and velocity components of a particle. p, f_i and ρ denote the pressure, external force, and density, respectively. It has been observed in the literature that the Arnold system displays periodic behavior as C or any other constants tend towards 0, and displays chaotic behavior when C is large. An example chaotic trajectory of the Arnold system is shown in Fig. 7. In addition to [121], other studies have used the Arnold system for path planning in 2D [160], [161] and 3D [152], [161] environments.

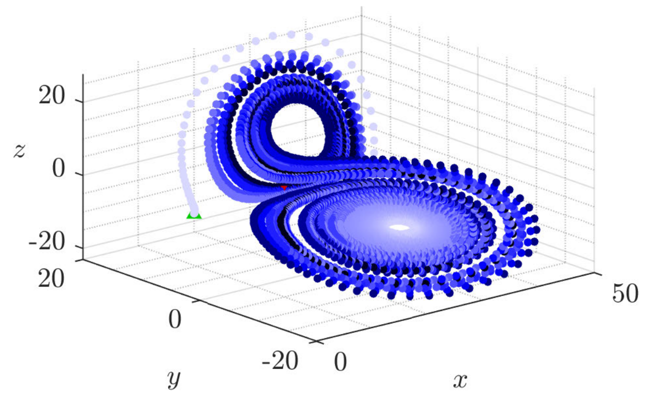


FIGURE 8. Lorenz attractor displaying chaotic behaviours under parameters $\sigma = 10$, $r = 28$, $b = 8/3$ [162].

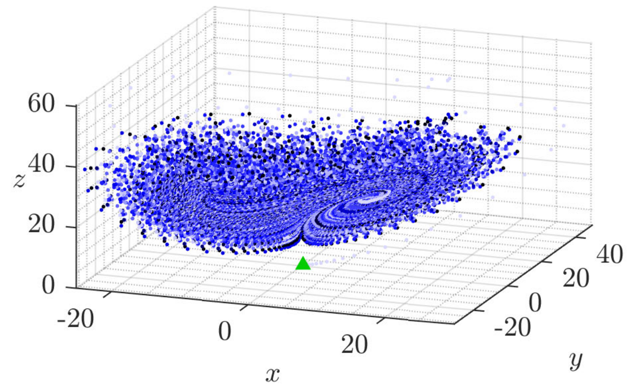


FIGURE 9. 3D Chen and Ueta attractor displaying rich chaotic behaviours with parameters $a = 35$, $b = 3$, $c = 28$.

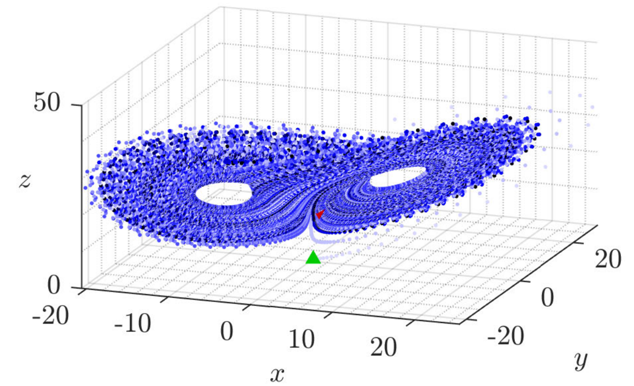


FIGURE 10. 3D Chen and Lü attractor displaying rich chaotic behaviours with parameters $a = 36$, $b = 3$, $c = 20$.

B. LORENZ SYSTEM

The Lorenz system is a set of nonlinear first-order differential equations originally published by American mathematician and meteorologist Edward Lorenz in 1963 [1]. Lorenz is a simplified mathematical model used to describe atmospheric convection. The equations modeling the Lorenz system are shown in (4) and are parameterized by σ, r, and b, which are all positive.

$$\begin{cases} \dot{x} = \sigma(-x + y) \\ \dot{y} = rx - y - xz \\ \dot{z} = -bz + xy \end{cases} \tag{4}$$

The equations describe the behavior of a two-dimensional fluid layer, subject to uniform heating from below and cooling from above. They detail the temporal evolution of three distinct properties: x corresponds to the rate of convection, y represents horizontal temperature variation, and z is associated with vertical temperature variation. The constants σ, r, and b are system parameters that are proportional to the Prandtl number, Rayleigh number, and specific physical dimensions of the layer. As observed in Fig. 8, Lorenz exhibits double-scroll attractors. Edward Lorenz realized that for $\sigma = 10$, $b = 8/3$, the system behaves chaotically whenever the Rayleigh number r exceeds the critical value of 24.74, meaning that almost all solutions appear to be sensitive to initial conditions. For this reason, studies [162], [163] interested in using the Lorenz system for chaotic path planning, considered the parameter tuple used to generate Fig. 8. For parameter values of $r < 1$, all solutions of the system will approach the equilibrium point located at the origin. However, as r gradually increases through 1, the eigenvalue situated at the origin becomes positive, causing the origin to transform into a saddle with a stable surface that spans two dimensions and an unstable curve. When $r = 1$, two equilibria are born and diverge from the origin as r increases. When $r > 1$, not all solutions converge towards the origin. However, it can be observed that these initial solutions, which are located at a considerable distance from the origin, gradually become closer to it over time [164].

C. CHEN SYSTEMS

The Chen system which is another nonlinear continuous first-order dynamical system has two variations that are used in the CPP literature: Chen and Ueta [155] and Chen and Lü [156] systems expressed by (5) and (6), respectively.

$$\begin{cases} \dot{x} = a(-x + y) \\ \dot{y} = (c - a)x - xz + cy \\ \dot{z} = -bz + xy \end{cases} \tag{5}$$

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = -xz + cy \\ \dot{z} = xy - bz \end{cases} \tag{6}$$

In the equations, x, y, and z are the state variables and a, b, and c are parameters that control the behavior of the systems. These parameters can be adjusted to produce different types of behavior, such as periodic or chaotic. In the standard formulation of the Chen and Ueta system, these values are set to $a = 35$, $b = 3$, and $c = 28$, which produce chaotic behavior shown in Fig. 9. The parameter values $a = 36$, $b = 3$, and $c = 20$ in Chen and Lü system will similarly produce chaotic behaviours displayed in Fig. 10. Similar to the Lorenz system, the Chen and Ueta system and Chen and

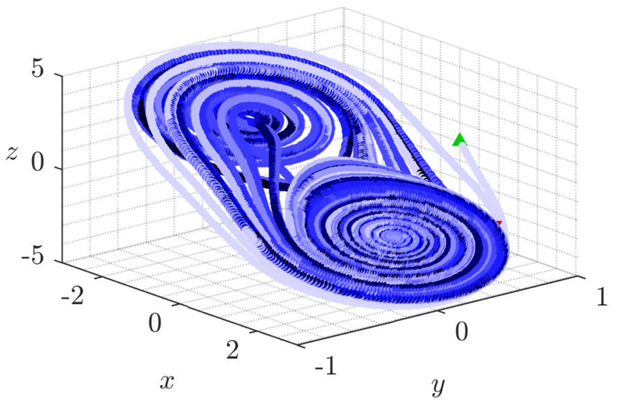


FIGURE 11. 3D Chua's circuit attractor displaying chaotic behaviours under parameters $\alpha = 10$, $\beta = 14.87$, $a = -1.27$, $b = -0.68$, $c = 1$.

Lü system have two attractors that will attract trajectories before expelling them although the attractors are not as co-planar as in the Lorenz system.

To the authors' knowledge, the only study that used Chen and Ueta for chaotic path planning has been [165] which combined Chen and Lorenz to enable monitoring of arbitrary points of interest using chaotic trajectories. A recent study [166] employed the Lü and Chen system for CPP tasks. The study focuses on ascertaining the optimal range for the parameter c while keeping the other two parameters fixed at $a = 36$ and $b = 3$. Upon investigating the system's chaotic properties using bifurcation diagrams and measures such as Lyapunov exponents, they determined the optimal range for c to be [20, 28]. Within this range, $c = 24$ corresponded to highest randomness and at the same time delivered the highest coverage rate.

D. CHUA CIRCUIT AND MULTI-SCROLL SYSTEMS

The Chua circuit, invented in 1983 by American engineer and computer scientist Chua [167], is a non-linear circuit that has been studied extensively for its application to chaos and chaotic path planning. By utilizing Kirchhoff's circuit laws to study the circuit, it is possible to precisely simulate the behavior of Chua's circuit through a set of three nonlinear ordinary differential equations (with state variables x, y, and z) that describe the voltages across capacitors, as well as the electric current in the inductor. Much like the Lorenz system, the Chua circuit exhibits double-scroll chaotic attractors and period-doubling bifurcation route to chaos as shown in Fig. 11. The differential equations governing the system are listed in (7).

$$\begin{cases} \dot{x} = \alpha(y - x - f(x)) \\ \dot{y} = x - y + z \\ \dot{z} = -\beta y \\ f(x) = bx + (a-b)(|x+c| - |x-c|)/2 \end{cases} \tag{7}$$

The nonlinear resistor's electrical characteristics are depicted by the function $f(x)$, where a, b, and c are geometric factors. The values of the circuit components determine the parameters α and β. Several studies have used either Chua system [168], [169] or multi-scroll systems inspired by the Chua circuit system [170] to generate chaotic paths. Chua-based multi-scroll chaotic systems resulted in higher performance compared with the double scroll Chua system. Unlike the double-scroll systems, path planners using the multi-scroll system do not generate unnecessarily dense trajectories [170]. In addition to Chua based systems, studies have utilized chaotic double-scroll and multi-scroll systems based on saturated function series to either directly generate deterministic yet unpredictable paths [171] or to create random number generators (RNG) that can produce trajectories with the statistical properties of a random sequence [169], [172]. Reference [173] used a combination of hyperjerk chaotic system and a modulo operator to determine the next direction for the robot. The input to modulo operator is the generated point by the hyperjerk system and the output is the next robot direction chosen out of the 4, or 8, possible options.

E. DISCRETE SYSTEMS

While the continuous systems discussed previously are defined by a differential equation, discrete systems utilize iterative functional mappings to generate the system states at each time index. The Logistic map shown in (8) and the Hénon map shown in (9) are the two popular systems used in chaotic path planners. The logistic map is a second-degree polynomial function that is commonly cited as a classic instance of how seemingly straightforward nonlinear dynamical equations can give rise to intricate chaotic behavior. The biologist Robert May brought the logistic map to public attention through the study [174] which presented the map as a discrete-time demographic model. In Eq. (8), $x_n \in [0, 1]$ is the ratio of existing population to the maximum possible population. The nonlinear difference equation reflects two key factors influencing population dynamics: reproduction and starvation. At low population sizes, reproduction causes the population to increase at a rate proportional to the current population. However, at higher densities, the growth rate decreases in response to the concept of carrying capacity, which represents the maximum population that the environment can support. This reduction in growth rate is proportional to the difference between the carrying capacity and the current population and is often referred to as density-dependent mortality or starvation. The parameter $r \in [0, 4]$ and its variations can significantly influence the system behavior. When $r \approx 3.56995$, the system starts to exhibit chaotic behavior, however, islands of stability still exist where the system shows non-chaotic behavior. While the chaos attractor for the one-dimensional Logistic map does not exist due to its inherent one-dimensional nature, we can readily observe the chaos attractor for the two-dimensional Hénon map system in Fig. 12. Similar to Logistic map, the Hénon map operates in

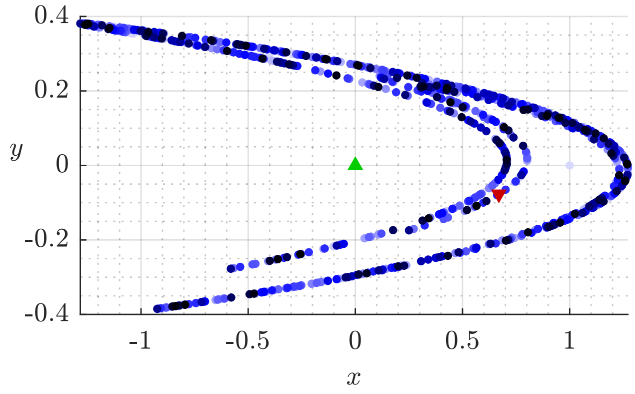


FIGURE 12. **Hénon map system attractor with parameters $a = 1.4$, $b = 0.3$.**

discrete time, meaning that it progresses through a sequence of distinct steps. In practical terms, the Hénon map takes a point situated on the plane, identified by its coordinates (x_n, y_n), and transforms it into a new point. The behavior of the Hénon map is determined by two specific parameters, known as a and b. In the standard formulation of the Hénon map, these values are set to $a = 1.4$ and $b = 0.3$, which produce the chaotic behavior. However, by modifying the values of a and b, the map can display various types of behavior such as intermittent dynamics or convergence to a periodic orbit. To gain an understanding of the different types of behavior exhibited by the map at various parameter values, one can examine its orbit diagram.

$$x_{n+1} = rx_n(1 - x_n) \tag{8}$$

$$\begin{cases} x_{n+1} = y_n + 1 - ax_n^2 \\ y_{n+1} = bx_n \end{cases} \tag{9}$$

When integrated into the differential wheeled robot's heading controller, both systems tend to drive the robot in only one direction which can be problematic in applications related to coverage of bounded environments. This can be addressed through applying a phase shift to the robot's angle and directing it to move in the opposite direction when it comes close to any boundary in the space. As an alternative to the robotic heading control, several studies have used the generated points by the Logistic map for direct or indirect coverage of the environment. In the direct method, the points generated by the chaotic system are used unaltered to cover the environment [175]. In contrast, the indirect method will apply a simple modulo operator to the Logistic map point sequence to determine the next direction for the robot to travel from a set of 4 or more possible choices. References [176], [177], [178], [179], and [180] use this indirect approach combined with a memory technique to improve the uniformity of the coverage relative to the direct approach. The memory technique allows the robot to recall the last few places visited and avoid revisiting those places in the near future. In addition to coverage tasks, these discrete systems are useful in generating direct chaotic paths to travel between two specific points as implemented in [160] and [162]. They can be used as well to produce unpredictable paths in the vicinity of a closed contour for boundary surveillance missions. The study [181] demonstrated the use of the Hénon map system to generate such paths.

The final discrete chaotic dynamical system discussed in this section is the Taylor-Chirikov, also known as the standard map. This map is used by [182] to directly generate trajectory points for the coverage task. The results showed similar desirable performance to the results of a RW path planner although the RW path planner results varied over different simulation runs. It should be noted that other discrete dynamical systems can be used for path planning, such as the Chebyshev map discussed in Section IV or Memristive Map [183], but the Logistic and Hénon maps are the most prevalent within the literature.

F. SYSTEMS COMPARISON

The difference between chaotic systems can be summarized in their coverage density and coverage extent which are directly related to their coverage time and rate. Some systems, e.g. Lorenz system [162], tend to generate locally dense trajectories and have difficulty extending their paths globally, particularly in large environments. Thus chaotic CPP using the Lorenz system tend to have relatively high coverage times. Other systems, e.g. the Arnold system [160], can better extend their paths to distant regions and generate less dense trajectories.

The level of randomness in trajectories is another distinguishing factor for chaotic path planners. To ensure that the generated trajectories exhibit the statistical characteristics of a random sequence, certain studies [177], [180], [184], [185], [186], [187] have designed pseudo-random bit generators (PRBGs) based on discrete systems such as the Logistic map. These PRBGs produce random sequences that pass various randomness tests and will be subsequently employed to generate motion commands in discrete directions.

Some chaotic path planning literatures [188] and [189] present comparisons of well-known chaotic equations. These comparisons are performed to establish which equations are better applicable to various testing conditions judging by their coverage rate. The chaotic equations are simulated with parameters that are carefully selected to produce the best possible coverage. Reference [189] presents a study on the trajectory of guiding signals of the Chua's attractor and the Arnold's attractor, and draws a conclusion that the trajectories from Chua's attractor cover more areas than that of Arnold's attractor. Reference [190] compares the Chua circuit, the Lorenz equation, and the Volos, Kyprianidis and Stouboulos (VKS) system [191] for their coverage rates. The results from [190] show that the VKS has better coverage rate due to the nature of double-scroll chaotic attractors produced by VKS. To date, the coverage rate has remained a widely used criterion for the determination of the selection chaotic equations for chaotic path planners although the selection is still highly dependent on the task and algorithm implementation. For example, if the task is to move between two specific points using a direct but unpredictable path, the Henon map or Logistic map have been shown to be good chaotic systems for these tasks [160], [162]. Alternatively, if the task is to completely search an area, the designers should consider the requirements of a coverage task such as the target coverage density and coverage extent. Some systems like Lorenz generate points with higher density but they have difficulty expanding the paths to distant areas, so they might not be efficient in terms of coverage time for covering a large environment unless we use some chaos control techniques discussed in Section IV.

Table 1 summarizes the performance of all the different methods/systems used by the studies. The Table presents various important parameters in chaotic path planning including coverage time (CT), total area covered (A_{TC}), robot detection area (A_D), robot velocity (v), coverage rate (CR), obstacle occupancy Obs_o, and the number of robots used for coverage (N_R). We also categorized the works based on the type of study they performed, namely, Simulation (Sim.) or Experiment (Exp.). Many of the studies included various versions of their method in different environments and among those, this survey selected the ones with the best performing parameters to present in the Table.

The studies on CPP, listed in Table 1, aimed at achieving one or more of the following objectives:

1) Developing techniques for manipulating chaos to enhance the coverage rate and improve the coverage uniformity.
2) Utilizing chaos synchronization techniques to facilitate multi-robot cooperative path planning tasks.
3) Focusing on covering more complex maps cluttered with obstacles, including large obstacles that have minimal impact on chaotic paths and densely packed small obstacles that can significantly affect CPP coverage.
4) Developing methods for manipulating chaotic systems to achieve rapid and complete coverage of large maps.
5) Presenting a new chaotic system for coverage path planning and analyzing its properties. These works explore various system characteristics, such as fundamental dynamic patterns, bifurcations, and routes to chaos, using analytical or computational methods.

To enable a more effective comparison between the methods/systems, this paper has defined the following dimensionless parameter that encompasses all the above-mentioned parameters in the coverage chaotic path planners:

$$PM = e^{-\alpha CR}\underbrace{(\frac{vt}{\sqrt{A_{TC}}})^{\beta}}_{\text{Parameter \#1}}\underbrace{(\frac{N_R\sqrt{A_D}}{\sqrt{A_{TC}}})^{\gamma}}_{\text{Parameter \#2}} \tag{10}$$

This measure penalizes methods with low coverage rates and adjusts the performance metric PM of multi-robot references by a factor of the number of robots involved (N_R). Lower PMs correspond to a better performance. When attempting to cover a given map, achieving a higher coverage rate will require more time compared to a lower coverage rate. For instance, it may take an infinite amount of time to fully cover 99% of a map, while covering 10% of a map may only take a second. In order to establish the exact nonlinear relationship between the coverage rate (CR) and its corresponding coverage time (t), this study calculated the time required to cover a range of rates from 0% to 100% for all the systems detailed in Section III. It then identified the model that fits the data for each system with a high goodness-of-fit ($R^2 > 92\%$). The results of this analysis are illustrated in Fig. 13. As observed, there is a strong exponential relationship between CR and t for these systems. The PM formula (10) captures this relationship. In addition, we divided other parameters in (10) into two parts: Parameter #1 which holds the greatest relevance to the robot itself, is taken to the power of β, and Parameter #2 which pertains mostly to the environment and has a power of γ. The values of α, β and γ depend on the objective of the comparison task. In this study, we use $\alpha = 0.42$, $\beta = 1$, and $\gamma = 1$. $\beta = 1$ and $\gamma = 1$ indicate the same significance for environment and robot related parameters. However, these values can be adjusted based on the desired weight for these properties. Assuming $\beta = 1$ and $\gamma = 1$, the value of α is determined by establishing the correlation between CR and PM by taking the natural logarithm of (11) from both sides which leads to set of equations described by (12). The values for parameters appearing in (12) were extracted from the papers listed in Table 1 which results in 28 useful data sets. The papers associated with incomplete data were excluded from the analysis. The variable y in Eqs.11 and 12 is an n-dimensional column vector. Solving Eq. 12 yields the value of α. It should be noted that the obtained α is only one of several possible solutions, given the limited size of the database. Therefore, the parameters can be adjusted according to the desired objectives.

$$\underbrace{(\frac{vt}{\sqrt{A_{TC}}})^{\beta}(\frac{N_R\sqrt{A_D}}{\sqrt{A_{TC}}})^{\gamma}}_{\text{y}} = e^{\alpha CR} \times PM \tag{11}$$

$$\begin{bmatrix} \ln y_1 \\ \ln y_2 \\ \vdots \\ \ln y_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 & CR_1 \\ 0 & 1 & \dots & 0 & CR_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & CR_n \end{bmatrix} \times \begin{bmatrix} \ln PM_1 \\ \ln PM_2 \\ \vdots \\ \ln PM_n \\ \alpha \end{bmatrix} \tag{12}$$

The last column of Table 1 displays the PM values calculated for each study using the parameter values mentioned above. This column excludes studies that did not provide all the necessary variable values for (10).

IV. CHAOS MANIPULATION

Although the equations described in Section III possess the desired characteristics of chaotic motion, they are mostly limited in functionality due to poor coverage of the search

TABLE 1. Performance comparison between chaotic coverage path planners.

Ref.	CDS	Method	Sim/Exp	CT (s)	A_{TC} (m^2)	A_D (m^2)	v (m/s)	$CR(\%)$	$Obs_o(\%)$	N_R	PM
1.[121]	Arnold	NA	Sim	8.00×10^3	4.73×10^2	0.06	1.000	90.00%	16.00%	1	2.84
	Arnold	NA	Sim	8.00×10^3	5.18×10^2	0.06	1.000	90.00%	8.00%	1	2.59
	Arnold	NA	Exp	5.00×10^2	2.92	0.06	0.120	90.00%	0	1	3.45
2.[124]	VKS	RNG	Sim	NA	1.60×10^4	NA	0.200	40.00%	0	1	NA
3.[159]	New	NA	Sim	NA	8.20×10^1	1.00	NA	82.00%	0	1	NA
4.[160]	Arnold	Partitioning, Orientation Control and Scaling	Sim	8.55×10^4	3.60×10^4	1.00	1.000	90.00%	0	1	1.63
	Arnold	Partitioning, Orientation Control and Scaling	Sim	6.57×10^4	2.12×10^4	1.00	1.000	90.00%	41.00%	1	2.12
	Arnold	Partitioning, Orientation Control and Scaling	Sim	2.00×10^4	3.60×10^4	16.00	1.000	90.00%	0	1	1.52
	Arnold	Partitioning, Orientation Control and Scaling	Sim	1.49×10^4	2.12×10^4	16.00	1.000	90.00%	41.00%	1	1.92
5.[162]	Lorenz-Hénon	Partitioning, Orientation Control	Sim	1.88×10^5	2.25×10^3	1.00	1.000	90.00%	0	1	57.25
6.[163]	Lorenz	Bounded Strategy	Sim	1.60×10^4	2.38×10^1	NA	NA	95.00%	0	1	NA
7.[166]	Chen	NA	Sim	1.00×10^4	0.91	NA	NA	91.25%	0	1	NA
8.[168]	VKS	NA	Sim	1.00×10^3	1.60×10^2	0.01	1.000	40.00%	0	1	0.53
9.[169]	Lorenz	RNG	Sim	5.91×10^4	7.59×10^2	NA	NA	84.33%	0	2	NA
10.[170]	Muti-scroll Chua	Flatness Control	Sim	8.00×10^4	8.99×10^3	0.01	3.000	89.86%	0	1	1.83
11.[171]	Double-scroll	RNG	Exp	NA	NA	NA	NA	NA	0	1	NA
12.[172]	Double-scroll	RNG	Sim	1.50×10^3	NA	NA	NA	80.00%	0	1	NA
13.[173]	Modify Hyperjerk	RNG	Sim	3.50×10^3	7.50×10^3	4.00	1.00	75.00%	0	1	1.36
14.[175]	Logistic Map	Trigonometric Transformations	Sim	1.50×10^4	9.90×10^3	1.00	1.000	99.00%	0	1	1.04
15.[177]	Logistic Map	PRBG	Sim	4.00×10^4	7.70×10^3	0.25	1.000	77.00%	0	1	1.88
	Logistic Map	PRBG	Sim	5.00×10^4	7.25×10^3	0.25	1.000	84.28%	14.00%	1	2.42
16.[180]	Logistic Map	PRBG-Memory Technique	Sim	1.00×10^4	7.99×10^3	4.00	1.000	79.93%	0	1	1.79
17.[183]	Memristive	NA	Sim	1.10×10^3	1.03×10^1	0.01	1.000	98.14%	0	1	7.07
18.[184]	Logistic Map	PRBG	Sim	2.00×10^5	1.13×10^5	NA	NA	90.00%	0	1	NA
19.[185]	Logistic Map	PRBG-Pheromone Model	Sim	3.50×10^4	7.82×10^3	NA	NA	78.24%	0	1	NA
20.[186]	Logistic Map	PRBG	Exp	NA	0.07	NA	NA	63.33%	0	1	NA
21.[187]	Logistic Map	PRBG	Exp	1.08×10^3	0.09	NA	NA	80.00%	0	1	NA
22.[188]	POSCH 7	Partitioning	Sim	1.00×10^3	8.74×10^1	1.00	0.120	87.37%	0	1	0.95
23.[189]	Chua	Partitioning	Sim	1.00×10^2	7.94	1.00	0.120	88.27%	0	1	1.04
24.[192]	Hénon	RNG	Sim	6.00×10^2	2.83×10^2	0.02	0.090	84.13%	0	1	2.06
	Hénon	RNG	Sim	5.00×10^1	2.88×10^2	0.02	0.090	85.83%	0	10	1.67
25.[193]	Logistic Map	NA	Sim	7.00×10^3	0.97	NA	0.150	96.62%	0	1	NA
26.[194]	Chebyshev	Trigonometric Transformations	Sim	2.00×10^4	NA	0.03	0.125	NA	0	1	NA
	Chebyshev	Trigonometric Transformations	Sim	1.00×10^4	NA	0.03	0.125	NA	25.00%	1	NA
	Chebyshev	Trigonometric Transformations	Sim	1.89×10^4	NA	0.03	0.125	NA	8.00%	1	NA
27.[195]	Chen	Synchronized	Sim	NA	9.00×10^1	0.01	1.000	90.00%	0	4	NA
	Chen	Synchronized	Sim	2.00×10^2	4.84×10^1	0.01	1.000	48.35%	0	4	1.35
	Chen	Synchronized	Sim	2.00×10^2	4.92×10^1	0.01	1.000	55.89%	12.00%	4	1.29
	Lorenz	Synchronized	Sim	6.94×10^2	9.00×10^1	0.01	1.000	90.00%	0	4	2.11
	Chen	Synchronized	Sim	6.66×10^2	9.00×10^1	0.01	1.000	90.00%	12.00%	4	2.30
28.[196]	Arnold	Synchronized	Sim	1.20×10^3	8.10	0.01	0.300	90.00%	0	2	6.09
	Arnold	Synchronized	Sim	9.00×10^2	6.75	0.01	0.300	90.00%	16.67%	2	5.48
29.[197]	Logistic Map	RNG	Exp	1.20×10^3	8.90×10^1	0.09	1.000	89.90%	0	1	2.77
30.[198]	Chua	Partitioning	Exp	1.00×10^3	7.98	NA	NA	88.72%	0	1	NA
31.[199]	Lorenz, Hamilton, Hyper	Synchronized	Exp	NA	NA	NA	1.000	NA	0	1	NA

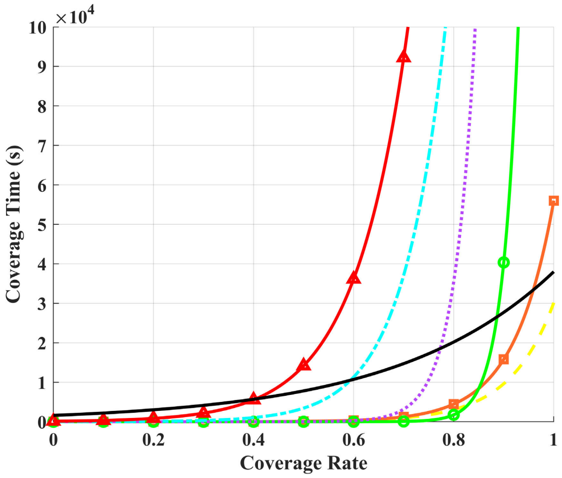


FIGURE 13. The relationship between Coverage Rate and Coverage Time for different Chaotic Systems. ——: Logistic System, - - - -: Arnold System, ...: Chua Circuit System, _._._.: Hénon System, —o—: Chen and Ueta System, —□—: Chen and Lü System, and —△—: Lorenz System.

environment. Consequently, several works in the literatures [160], [162], [165], [170], [175], [194], [195], [200], [201], and [202] have innovated numerous means of improving the functionality and chaotic properties in processes collectively known as chaos manipulation. Chaos manipulation involves the adjustments of chaotic algorithms by methods such as: 1) combining two or more chaos equations; 2) transformations of the chaotic and/or environment states; and 3) adding of points of interest into or partitioning the robot search environment. These manipulations are done to increase the performance of the chaotic CPP with regards to the coverage time and the path density. Various methods of chaos manipulation are described in the following subsections.

A. ARCCOSINE AND ARCSINE TRANSFORMATION

Reference [194] describes the use of trigonometric transformations such as arccosine and arcsine transformations to increase the coverage rate and coverage distribution of the Chebyshev map for special missions. The Chebyshev map (13) is a simple dynamical system that bounds the chaotic state x_n in the range -1 to 1.

$$x_{n+1} = f(x_n) = \cos(k \cdot \arccos x_n) \tag{13}$$

where k denotes the order of the map. The Chebyshev map displays uneven distribution at the extreme values thereby implying poorly dispersed coverage. Reference [194] uses arccosine and arcsine transformations to induce higher topological transitivity for more even coverage of the range. The transformations of the Chebyshev maps and the corresponding plots are shown in Fig. 14. Reference [175] also makes use of these transformations to solve a similar problem of concentrated coverage at the boundaries when using the Logistic map system for the coverage task. However, it combines the arcsine and arccosine transformation to create an improved path planner with better uniformity and randomness using less iterations than the untransformed Logistic map system.

B. PLANNED SUB-GOALS, POINTS OF INTEREST (POIS), AND AFFINE TRANSFORMATIONS

Another method of chaos manipulation involves the use of planned sub-goals or points of interest. Reference [200] proposes a fusion iteration strategy using affine transformations to impart the standard chaotic algorithm onto the robot controller. The chaotic trajectory produced by the robot controller in this strategy has intermediate adjacent points with large distances between them, which are therefore difficult for the robot to track. Consequently, they created an algorithm that enhances the robot's navigation. The algorithm creates sub-goals between adjacent points, divides the environment into large regions, and further divides those into small grids. To realize the fusion iteration strategy, the code uses functions to judge if each small grid has been totally covered, and, if not, starts up the small grid iteration cycles till complete coverage of the grid is achieved, before moving on to an adjacent small grid using the large iteration cycle. Using this manipulation method, the algorithm is able to reduce the distance between the adjacent points. However, the coverage rate and time of the manipulated chaotic CPP is not indicated in [200].

Reference [165] discusses another robot navigation enhancement strategy using points of interest (POIs) rather than planned sub-goals. In some specialized chaotic path planning applications, there is a need to monitor specific POIs; therefore, the goal of [165] is to create a chaotic system for monitoring an arbitrary number of specified locations, consequently proposing a methodology to impart both real and synthetic POIs to the robot controller. The real POIs are chosen as targets/goals to be monitored by the robot, and synthetic POIs are chosen strictly to induce more unpredictability of the path. After the POIs are established, the next step involves the formation of cells to contain the points of interest. Consequently, a schedule is created to impart the robots planned time in each cell, and then the local Cartesian coordinate system inside each cell is established. Finally, trajectory segments are generated inside each cell. The chaotic trajectory used by this study comes from a combination of Lorenz and Chen equations. The study used affine transformations to transfer the twin equilibrium points of Lorenz and Chen to POIs and thereby transfer and form the chaotic systems' trajectories around these points. The affine transformation includes a rotation and a scaling transformation. The simulation of the derived system shows dense trajectories around the points of interest, thereby achieving the goal of this study which is demonstrated in Fig. 15. However, unlike other studies such as [160], [162], and [163], this work did not use a dynamical system coordinate to perturb the robot's orientation and impart chaos into the robot's controller. Instead, the two coordinates of the mobile robot are just a simple transformation of coordinates of the dynamical system. As a result, the appearance of

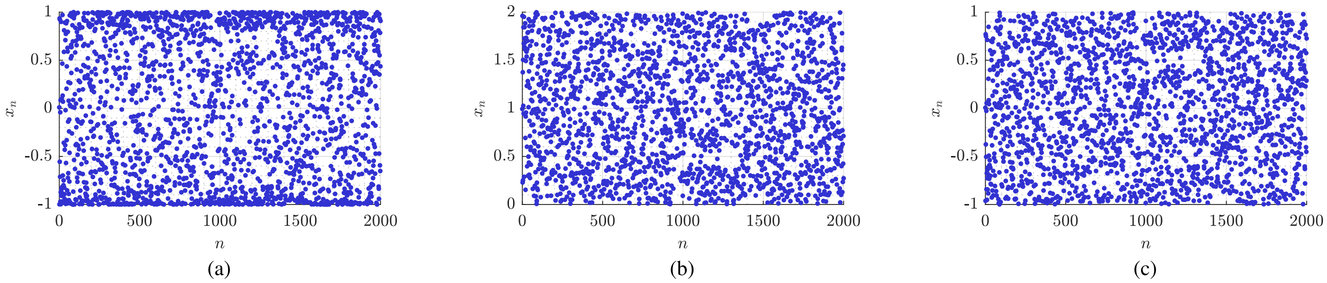


FIGURE 14. **Trajectory distribution of Chebyshev map with $k = 6$ showing: (a) unequal distribution at both ends, (b) arcsine transformation with more dispersed distribution, (b) arccosine transformation with more dispersed distribution.**

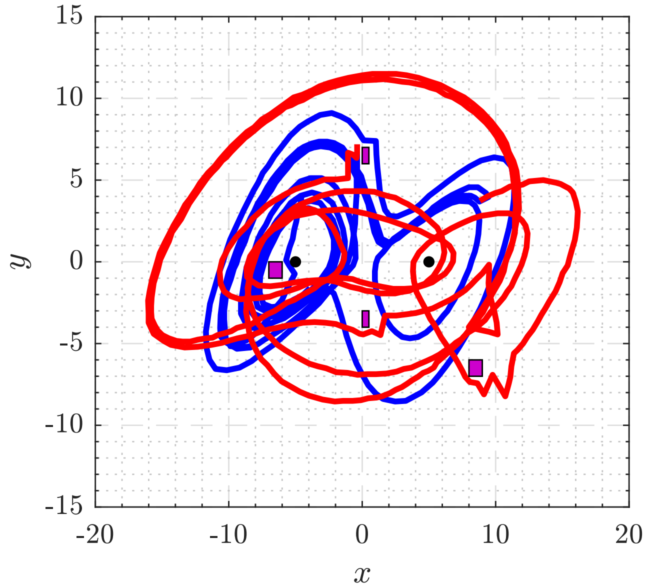


FIGURE 15. **Lorenz and Chen system monitoring two points in an environment with obstacles from [165]. Obstacles in the environment are shown as rectangular blocks with POIs marked by the black circles placed at $(-5, 0)$ and $(5, 0)$. Figure was adapted with permission from [165].**

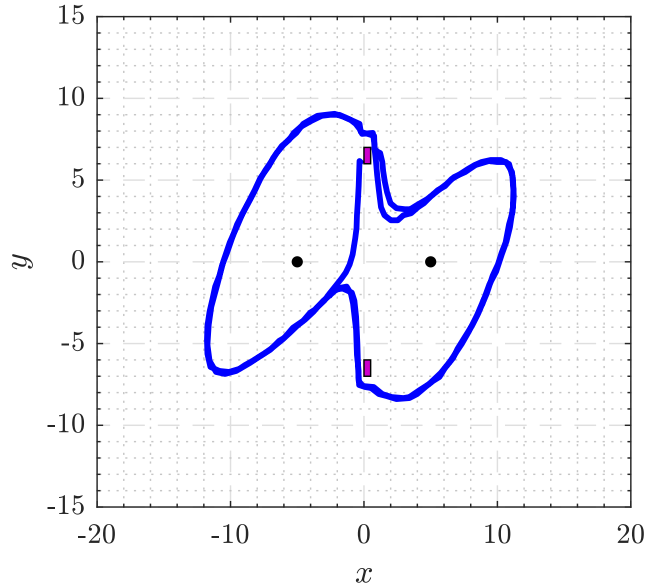


FIGURE 16. **Lorenz chaotic trajectory forced to become periodic [165]. Obstacles in the environment are shown as rectangular blocks with POIs marked by the black circles placed at $(-5, 0)$ and $(5, 0)$. Figure was adapted with permission from [165].**

obstacles in the path might degrade the chaotic trajectories and turn them into periodic paths as illustrated in Fig. 16. To solve this issue, [165] used synthetic POIs to re-establish the chaotic behavior of the Lorenz equation when obstacles exist on the trajectory.

C. CHAOS SYNCHRONIZATION

The next chaos manipulation method to be explored is the use of chaos synchronization to develop more complex chaotic systems in a multi-robot application. Chaos synchronization is used to harmonize and create identical trajectories to achieve effective cooperation in a multi-robot chaotic system. This process involves forcing two or more chaotic systems to follow identical paths by ''locking'' all systems to one system i.e., the drive system. Chaotic systems are known to resist synchronization due to their positive Lyapunov exponents; nevertheless, when both systems share information in a very particular manner, synchronization is possible [201], [203]. This is depicted in [195] and [196], where the robot's chaotic motions are implemented in such a way that there is a leader robot imparted with the chaos equations, and the other robots are synchronized to it. The four robots used in [195] all have the same kinematic conditions but different initial positions, dispersed to all four edges of the workspace. The robots traverse the workspace to accomplish important targets previously imparted on the controller, while the synchronization and obstacle avoidance control logic keeps the robots from colliding with each other or with the workspace boundary. The synchronization method used by [195] shows that chaos synchronization achieves better coverage in a multi-robot system than unsynchronized robots or random-walk robots. The study also showed that after 200 iterations, the synchronized system achieves 48.35% coverage as compared to 41.55% and 42.10% coverage in an unsynchronized and a random-walk coverage planning, respectively, under the same kinematic and workspace conditions.

D. PARTITIONING, ORIENTATION CONTROL, AND SCALING

The map partitioning method, known as map zoning, first proposed by [160] and [162], aimed to increase the uniformity and reduce the repetitiveness of the robot's path, and thereby reduce the coverage time. To achieve these objectives, the methods involve dividing the environment into both small and large cells to track the coverage in two different levels. The small cells, each corresponding to a region no larger than the robot's sensing range (SR), monitor the coverage status of these small areas to ensure the desired environment coverage. The large grid cells, or map partitions, aim to track the status of coverage of large portions of the environment. This utilization of small and large cells is reminiscent of the methods, particularly STC, discussed in Section II. The methods in [160] and [162] use a combination of a continuous and a discrete chaotic system for the chaotic path planner. The continuous system is used to cover the smaller cells with a dynamical state being used to directly control the robot's steering angle. The discrete system is used to move the robot to a new large cell as soon as repetitive coverage is detected. Overall, the method helps the robot to expand trajectories to remote regions.

The orientation control is another manipulation technique proposed by [160] and [162]. The method allows to dramatically change the robot's direction and drive it to cover more of adjacent grid cells, thereby preventing dense local trajectories. This is done through switching the dynamical system coordinate that determines or perturbs the robot's heading angle. Finally, the last method proposed by [160] has been the system scaling. The technique applies a factor to scale the robot's trajectories, allowing the algorithm to adjust the coverage density based on the robot's detection range. At the same time, the method provides the capability to vary the coverage extent based on the environment size. Overall, the technique increases the adaptability of chaotic systems to changes in the sensing range and environment size. Lack of adaptability to different robot/environment conditions has been a practical limitation of chaotic path planners. The combination of the above-mentioned three techniques helped to significantly reduce the repetitive coverage and the coverage time, making the performance comparable to that of optimal CPP methods discussed in Section II. However, the robot requires localization capabilities for successful implementation of the map zoning and orientation control techniques which deviate from purely chaotic trajectories in [121] or the random-walk methods.

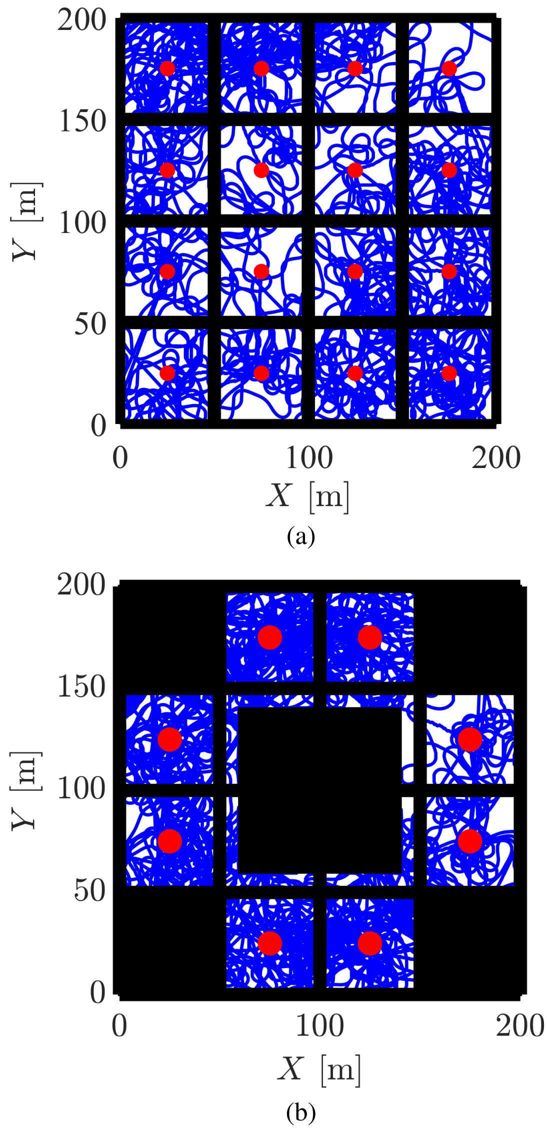


FIGURE 17. Arnold with all three chaos control techniques from [160] at robot sensing ranges of (a) 4 meters and (b) 1 meter. The black squares are obstacles and red circles indicate the zone midpoints. Figures were adapted with permission from [160].

V. OTHER APPLICATIONS OF CHAOTIC SYSTEMS IN ROBOTIC PATH PLANNING

In addition to their wide applications in CPP algorithms, chaotic equations have been used as well for finding the optimal collision free path between two points. The study [204] introduces an improved Moth-Flame Optimization algorithm called Opposition-Based Learning and Cauchy Mutation Moth-Flame Optimization (OLTC-MFO), designed for intelligent point-to-point route planning for UAVs. While the Moth-Flame Optimization Algorithm (MFO) shows promises in the field of intelligent optimization for path planning, it struggles to handle real-world disturbances, which can compromise the accuracy of the resulting route. Within the optimization landscape, chaotic mapping techniques have become increasingly popular in intelligent algorithms, due to their unique characteristics such as ergodicity, randomness, and nonlinearity. The study has used two well-known variants of chaotic maps, namely, the Logistic map and the Tent map. The former excels in global ergodicity but falls short in local searches, whereas the latter compensates for these limitations with its strong disturbance capabilities. The paper proposes a fusion of these maps, called Logistic-Tent Chaos Mapping (LT), to enhance the MFO algorithm's global search abilities by leveraging the potential of Logistic and Tent maps in generating chaos-driven disturbances to the flame position. It also incorporates a Cauchy mutation operator and a probability operator to perturb the optimal flame position. These modifications allow the algorithm to accept the current optimal solution with a certain probability, while also providing a mechanism to escape local optima. Reference [11] used a chaos modified genetic algorithm (GA) to find an optimal path to send goods from a warehouse to a customer while avoiding static obstacles in the environment. The optimal path in this task is calculated by finding the shortest distance between the point sequences of an abstract 3D model of the environment. The genetic algorithm uses the theory of genetics and evolution to find optimal solutions to path planning by varying and patching the best performing paths from a collection of candidate paths. However when GA is applied to nonlinear systems, the algorithm presents various shortcomings such as lack of a global convergence guarantee. A chaotic dynamical system, a Logistic map in [11], is combined with the GA because its ergodic tendency helps the GA avoid the path search from attraction towards local optimum paths rather than the true global optimum path. The improved chaotic GA is proven to be effective in large environments that necessitate both a global and a local search.

Another example of using chaos to find an obstacle-free path between two points is [205] that introduced a path planning equation. This continuous time differential equation uses chaos and the virtual force field (VFF) method to find a collision-free path while considering robot's dynamics,

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + d_1 \sin \omega t \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + d_2 \begin{bmatrix} \dot{x}^3 \\ \dot{y}^3 \end{bmatrix} = F_t + F_a + F_r \tag{14}$$

where m is the mass, d_1 and d_2 are parameters influencing the maximum magnitude of time-varying equilibrium points, F_t is the total virtual force, F_a is a virtual attractive force, and F_r is the resultant repulsive force of all virtual repulsive forces in the circular active window. The attractive force pulls the robot towards the goal while the repulsive force pulls it away from obstacles and boundaries in the environment. This force can become zero and remain zero if the robot is trapped in the local minima. This results in zero velocity for the robot and prevents it from reaching the goal position. To solve this problem, a nonlinear friction, the second and third components in the right-side of (14), is introduced into a chaotic neuron. The resulting instability in the equilibrium point of null velocity enables the robot to escape the local minima and to continue its path toward the goal.

Several other examples of chaotic dynamical systems being used for modifying existing shortest path algorithms are prevalent within the literature: 1) Chaos optimized artificial potential field [206]; 2) chaotic particle swarm optimization [207], [208], [209], [210]; 3) chaotic artificial bee colony optimization [202], [211]; and 4) chaotic Cuckoo search methods [205]. Reference [182] also proposed using a combination of Logistic mapping and Ulam-von Neumann mapping to solve a nonlinear constraint optimization problem to find an optimal path for a mobile robot. Chaotic systems may also be employed in path planning for more tailored purposes such as boundary surveillance [212]. Reference [181] proposes a path planning methodology using the Hénon discrete system to focus on boundary surveillance. The objective is to design a computationally simple chaotic system that exhibits chaotic trajectories when in proximity of any arbitrarily chosen closed contour. The methodology involves first selecting a closed contour and choosing the robot evolution vicinity in a 2D space, and then using affine transformations to impart the Hénon chaotic system onto the chosen path to construct the new chaotic trajectory. This new trajectory roughly follows the boundaries being surveyed but any adversarial agent hoping to sneak past the patrolling robot would have a difficult time predicting the position and trajectory of the patrolling robot.

VI. OBSTACLE AVOIDANCE IN CHAOTIC SYSTEM

Obstacle avoidance constitutes an important part of the path planning literature as an addition to finding a path from the starting point to an end point, it is also important for the robot to avoid colliding with any obstacles or boundaries in the space. Several methods of obstacle avoidance have been explored for various path planning algorithms and some of these methods involve the use of a relatively large number of sensors [109], [123]. In addition to localization or target detection, multi-purpose sensors can also alert the vehicle or robot of its closeness to any object, thereby informing the vehicle or robot of the need to change the path to avoid the obstacles.

Sensors for obstacle avoidance are also used in chaotic path planners. The sensor configuration of the robot determines the actions that the robot takes after sensing an obstacle in its path. These avoidance actions hold significant importance as they showcase the robot's level of autonomy. Robots with limited autonomy may be forced to stop and wait for direction from a human operator once within a range of any object, while more autonomous robots will realize the presence of an obstacle and independently decide on a suitable course of action. In [197], these two approaches are used as a means of obstacle avoidance for a robot equipped with sensors. Once the robot is less than 20 cm away from an obstacle, the robot either stops and requires a command, or turns right and continue the motion. Although the latter option increases the complexity of the robot's motion, the former option shows a slightly better coverage rate. Reference [165] also use a similar method of obstacle avoidance, whereby the robot uses ultrasonic sensors to detect the distance to an obstacle. This method is similar to the method in [197] where the robot would rotate left until the obstacle is out of view before continuing on the chaotic path if the obstacle was found to be too close to the chaotic path.

Mirror mapping is another popular method used in the literature. In this method [195], [199], [213], the object moves away from a boundary with the reflection angle equal to the incident angle as though it is reflected by the boundary, as shown in Fig. 18. Mirror mapping allows for arbitrary turning angle from obstacles rather than the prescribed turning angle methods of [165] and [197].

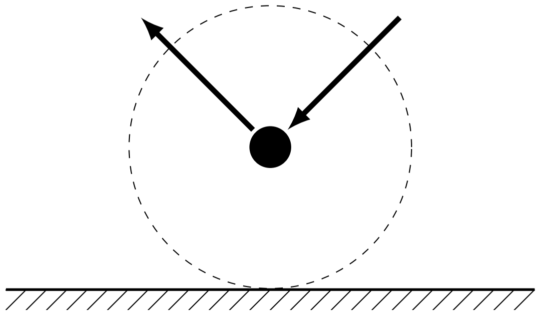


FIGURE 18. Obstacle avoidance using mirror mapping technique (filled circle: robot; dashed circle: robot sensing range).

All the methods discussed in this section rely on relatively simple implementations within the chaotic path planners which allows for minimal alterations to the original CPP algorithms. However, they do not necessarily correspond to smooth trajectories and often lead to many instances of sharp turns. To address these issues, a recent study [161] proposed a new obstacle avoidance technique which uses a quadtree and a cost function to generate smooth and continuous trajectories around obstacles in real-time applications. The quadtree is created from descritized map data, enabling fast retrieval of coordinates within the occupancy grid map that represent the free space. The cost function is then employed to find a favorable replacement coordinate for non-viable trajectory points, using the information provided by quadtree.

VII. ENDURING LIMITATIONS OF CHAOTIC PATH PLANNING AND FUTURE WORK

One of the multiple reasons that make chaotic path planning worth studying is the potential solutions it can provide to myriads of robotic engineering problems that desire unpredictability and autonomy. Chaotic equations provide the unpredictability feature by nature but autonomy for chaotic CPP systems has been rarely discussed in the related literature. In some applications, there is a need for the interference of a human operator, which undermines the ability for the robot to achieve full autonomy. Additionally, although the coverage area is often discussed in chaotic literature and there are various methods created to manipulate and increase the area covered by the robot, there are comparatively fewer studies that discuss the coverage time, i.e. the time required to achieve this coverage. Consequently, the autonomy and coverage time of the chaotic mobile robot are aspects of path planning that pose the largest limitations to chaotic path planning as discussed in the following subsections.

A. AUTONOMY AND REAL-LIFE IMPLEMENTATIONS

In highly complex applications of chaotic path planning such as some military applications, there is a need for robots to be highly autonomous and intelligent. This implies the ability of a robot to navigate an environment without needing the influence of a human operator. In much of the chaotic path planning literature, the autonomy of the robot is rarely taken into consideration. Moreover, the methods of chaos manipulation used by some studies induce a reduction in the robot's level of autonomy. For instance, the chaos manipulation methods used by [165] made it necessary to compute points of interest to ease the robot navigation. They also made it important that a human controller creates a schedule of time that the robot needs to spend in each cell for the manipulated chaotic algorithm to be fully functional. A fully autonomous machine implies that the robot can solve problems encountered during navigation without the presence of a human operator. Semi-autonomous robots are largely incapable of handling uncertainties associated with real-life situations.

Ensuring autonomy and desirable performance would require implementing the chaotic methods on physical mobile robots or aerial vehicles and in realistic environments that has been rarely done by the literature. The studies have been mostly limited to examining the chaotic algorithms in simulated environments using simple differential wheeled robots. Use of real robots and sensors for chaotic coverage will require formulating a more sophisticated obstacle avoidance approach since the current common use mirror-mapping technique causes sharp angles in the paths close to boundaries/obstacles that might be difficult to follow by a physical robot. Moreover, the focus of literature has been so far on avoiding static obstacles within an environment with a known map. More research should be focused towards environments with dynamic obstacles and uncertain maps, particularly when developing new obstacle avoidance techniques.

There are also additional challenges towards real-world implementations such as simulating the continuous chaotic systems discussed in Section III. As the systems are chaotic, the system states evolution through time needs to be solved numerically which leads to chaotic degradation [214]. Chaotic degradation is the rise in error of the simulated trajectory in chaotic dynamical system from a true trajectory given any initial state. This is inevitable due to the numerical methods used and the chaotic nature of the system with positive MLE. The magnitude of the trajectory error is dependent on the selection of the chaotic system, the numerical scheme, and the time step selected. Given the robotic platform specifications, this may limit the choices available to the system designer but the influence of chaotic degradation ultimately depends on the task being undertaken.

B. COVERAGE TIME

A very important criteria for chaotic path planning analysis is the coverage time which has been overlooked in the chaotic literature. Ideally, a chaotic path planning method can achieve full coverage if allowed to run infinitely. However, it is unreasonable to expect or desire an infinite amount of time for the robot to cover the environment for real-life path planning applications. Thus the coverage time should be considered in addition to considerations for the coverage of any area in the environment. The very few chaotic path planning papers discussing coverage time display outcomes that may be impractical in real-life applications. In [121], it takes more than two hours for the robot to achieve what is presented as full coverage of a 25 m × 25 m space moving at a velocity of 1 meter per second on a chaotic path. Repetitive coverage is one of the factors contributing to poor coverage time. The results of [197] show that the robot has covered some cells 18 times, while there are cells that remain untouched. Only recently has [160], [162] paid particular attention to the coverage time and developed methods to enhance it while comparing the results with the time taken by an optimal simple planner to cover the same environment. Numerous previous works have concentrated on simulated environments while simultaneously advocating for more research and exploration into enhancing real world applications and further reduce the coverage time. Using advanced obstacle methods appears to be the most practical avenue of progress towards achieving this goal as advanced methods would reduce repetitive coverage near the obstacles or boundaries.

C. FUTURE WORK

The future research should prioritize enhancing the real-life applicability and performance of the chaotic path planning algorithms and further integrate them with physical robotic systems. This entails addressing real-world constraints such as time limitations, sensing range restrictions, and vehicle motion constraints. Additionally, it involves developing methods for fast real-time computation of sensory data and making instantaneous decisions regarding system parameters and specifications.

The integration of machine learning (ML) into CPP algorithms can facilitate real-life applications through enhancing performance across various aspects. These include improving coverage uniformity, reducing coverage time, increasing coverage rate, enhancing adaptability to unfamiliar cluttered environments, and ensuring scalability to maps of different sizes and shapes. One promising application area is utilizing ML to guide CPP algorithms in selecting suitable chaotic systems or combinations thereof and determining optimal parameters based on task requirements and environmental characteristics. This aids in achieving the desired coverage rate, time, or level of unpredictability. These ML algorithms can also be trained to predict optimal system parameters to create a balance between overall coverage performance and level of unpredictability of the trajectories. Reinforcement learning (RL) algorithms offer a means to enhance the real-time performance of chaotic agents. RL can be harnessed to train agents, enabling them to navigate cluttered environments smoothly while maintaining kinematic efficiency and mitigating disruptions to chaotic paths and preserving their smoothness in the presence of both static and dynamic obstacles. RL can as well enable the agents to make real-time decisions on system parameters to better disperse the trajectory points.

VIII. CONCLUSION

The simple stated primary goal of chaotic coverage path planning research is to create an autonomous mobile robot chaotic system that can fully cover any environment containing static and/or dynamic obstacles under a given realistic operational time constraint. Chaos has a unique power in realizing unpredictable and, at the same time, deterministic path planning, which is otherwise impossible using other types of path planning methods. After surveying the research done on several chaotic equations including Arnold, Lorenz, Chen, etc., it is fair to conclude that there has not been much progress with using these chaotic dynamical systems for a real-life robot undertaking a coverage task. Although manipulation of these chaotic systems often leads to favourable coverage outcomes, e.g. in the trigonometric transformation of Chebyshev maps, it is difficult to conclude that these favourable outcomes can be realistically applied to real-life robotic applications because they were not evaluated based on real-life constraints such as including time limitations, sensing range limitations, and vehicle motion constraints. This illuminates a huge research gap that needs to be explored in more depth. Furthermore, since multi-robots offer more versatility in real-world chaotic coverage and task assignment applications, more testing and research on applying any progress made in chaotic path planning to multi-robot systems must be done. To this end, the fully autonomous application of chaotic path planning calls for more research to address challenges associated with autonomy and coverage time. Realizing intelligent chaotic robots will enable effective online coverage of fully unknown and uncertain environments desired in many real-life applications, particularly in applications involving adversarial agents.

REFERENCES

[1] E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.*, vol. 20, no. 2, pp. 130–141, Mar. 1963.
[2] V. Bjerknes, *Dynamic Meteorology and Hydrography*. New York, NY, USA: Carnegie, 1910.
[3] D. Orrell, "Role of the metric in forecast error growth: How chaotic is the weather?" *Tellus A: Dyn. Meteorol. Oceanogr.*, vol. 54, no. 4, p. 350, Jan. 2002.
[4] E. N. Lorenz and K. Haman, "The essence of chaos," *Pure Appl. Geophysics*, vol. 147, no. 3, pp. 598–599, 1996.
[5] J. Slingo and T. Palmer, "Uncertainty in weather and climate prediction," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 369, no. 1956, pp. 4751–4767, 2011.
[6] K. T. Alligood, T. D. Sauer, and J. A. Yorke, *Chaos*. Cham, Switzerland: Springer, 1996.
[7] B. B. Ferreira, A. S. de Paula, and M. A. Savi, "Chaos control applied to heart rhythm dynamics," *Chaos, Solitons Fractals*, vol. 44, no. 8, pp. 587–599, Aug. 2011.
[8] C. Haigh, "Using chaos theory: The implications for nursing," *J. Adv. Nursing*, vol. 37, no. 5, pp. 462–469, Mar. 2002.
[9] S. Ayers, "The application of chaos theory to psychology," *Theory Psychol.*, vol. 7, no. 3, pp. 373–398, Jun. 1997.
[10] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Cham, Switzerland: Springer, 2016.
[11] G. Xusheng, G. Wenming, and H. Jun, "Robot path planning based on genetic and chaotic optimization algorithm," in *Proc. 5th Int. Conf. Comput., Automat. Power Electron. (CAPE)*. London, U.K.: Francis Academic Press, 2017, pp. 1–6, doi: 10.25236/cape.2017.023.

[12] O. Wahhab and A. Al-Araji, "Path planning and control strategy design for mobile robot based on hybrid swarm optimization algorithm," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 3, pp. 565–579, Jun. 2021.

[13] A. Ait-Saadi, Y. Meraihi, A. Soukane, S. Yahia, A. Ramdane-Cherif, and A. B. Gabis, "An enhanced African vulture optimization algorithm for solving the unmanned aerial vehicles path planning problem," *Comput. Electr. Eng.*, vol. 110, Sep. 2023, Art. no. 108802. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790623002264

[14] J. Yu, G. Liu, J. Xu, Z. Zhao, Z. Chen, M. Yang, X. Wang, and Y. Bai, "A hybrid multi-target path planning algorithm for unmanned cruise ship in an unknown obstacle environment," *Sensors*, vol. 22, no. 7, p. 2429, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/7/2429

[15] D. V. Lyridis, "An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints," *Ocean Eng.*, vol. 241, Dec. 2021, Art. no. 109890. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801821012385

[16] M. S. Gharajeh and H. B. Jond, "An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system," *Ain Shams Eng. J.*, vol. 13, no. 1, Jan. 2022, Art. no. 101491. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2090447921002276

[17] Q. Wang, J. Guo, Z. Wang, E. Tahchi, X. Wang, B. Moran, and M. Zukerman, "Cost-effective path planning for submarine cable network extension," *IEEE Access*, vol. 7, pp. 61883–61895, 2019.

[18] M. Xi, J. Yang, J. Wen, H. Liu, Y. Li, and H. H. Song, "Comprehensive ocean information-enabled AUV path planning via reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17440–17451, Sep. 2022.

[19] Y. Ma, W. Feng, Z. Mao, H. Li, and X. Meng, "Path planning of UUV based on HQPSO algorithm with considering the navigation error," *Ocean Eng.*, vol. 244, Jan. 2022, Art. no. 110048. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801821013780

[20] H. Huang, P. Huang, S. Zhong, T. Long, S. Wang, E. Qiang, Y. Zhong, and L. He, "Dynamic path planning based on improved D* algorithms of Gaode map," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Mar. 2019, pp. 1121–1124.

[21] P. M. Kolahi and M. Nazemizadeh, "Nonlinear dynamic modeling of tractor-trailer mobile robots with consideration of wheels inertia and their optimal point-to point path planning," *Meccanica*, vol. 58, no. 1, pp. 245–253, Jan. 2023.

[22] K. R. Hong, R. H. Kim, T. S. Kim, and J. S. Kim, "Minimum dose path planning for facility inspection based on the discrete rao-combined ABC algorithm in radioactive environments with obstacles," *Nucl. Sci. Techn.*, vol. 34, no. 4, pp. 1–15, Apr. 2023.

[23] N. A. K. Zghair and A. S. Al-Araji, "Intelligent hybrid path planning algorithms for autonomous mobile robots," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 5, pp. 309–325, 2022.

[24] Y. Liu and X. Li, "A hybrid mobile robot path planning scheme based on modified gray wolf optimization and situation assessment," *J. Robot.*, vol. 2022, pp. 1–9, Feb. 2022.

[25] X. Xie, Z. Tang, and J. Cai, "The multi-objective inspection path-planning in radioactive environment based on an improved ant colony optimization algorithm," *Prog. Nucl. Energy*, vol. 144, Feb. 2022, Art. no. 104076. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0149197021004303

[26] S. Dian, J. Zhong, B. Guo, J. Liu, and R. Guo, "A smooth path planning method for mobile robot using a BES-incorporated modified QPSO algorithm," *Exp. Syst. Appl.*, vol. 208, Dec. 2022, Art. no. 118256. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422014002

[27] S. Aslan and T. Erkin, "A multi-population immune plasma algorithm for path planning of unmanned combat aerial vehicle," *Adv. Eng. Informat.*, vol. 55, Jan. 2023, Art. no. 101829. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034622002877

[28] N. Du, Y. Zhou, W. Deng, and Q. Luo, "Improved chimp optimization algorithm for three-dimensional path planning problem," *Multimedia Tools Appl.*, vol. 81, no. 19, pp. 27397–27422, Aug. 2022.

[29] M. Zhou, Z. Wang, J. Wang, and Z. Dong, "A hybrid path planning and formation control strategy of multi-robots in a dynamic environment," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 26, no. 3, pp. 342–354, May 2022.

[30] C. Zhou, S. Gu, Y. Wen, Z. Du, C. Xiao, L. Huang, and M. Zhu, "The review unmanned surface vehicle path planning: Based on multi-modality constraint," *Ocean Eng.*, vol. 200, Mar. 2020, Art. no. 107043. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801820301177

[31] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technol.*, vol. 15, no. 4, pp. 582–606, Aug. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214914718305130

[32] T. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr., "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, p. 4, 2019. [Online]. Available: https://www.mdpi.com/2504-446X/3/1/4

[33] H. Wu, Y. Zhang, L. Huang, J. Zhang, Z. Luan, W. Zhao, and F. Chen, "Research on vehicle obstacle avoidance path planning based on APF-PSO," in *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 237, no. 6, pp. 1391–1405, doi: 10.1177/09544070221088364.

[34] Y. Zhang, B. Ou, Y. Xu, and C. Dai, "Path planning of mobile robots based on genetic algorithm," in *Proc. 8th Int. Conf. Cloud Comput. Big Data Analytics (ICCCBDA)*, Apr. 2023, pp. 501–505.

[35] H. Zhang, B. Xin, L.-H. Dou, J. Chen, and K. Hirota, "A review of cooperative path planning of an unmanned aerial vehicle group," *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, no. 12, pp. 1671–1694, Dec. 2020.

[36] Z. Wei, M. Zhu, N. Zhang, L. Wang, Y. Zou, Z. Meng, H. Wu, and Z. Feng, "UAV-assisted data collection for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15460–15483, Sep. 2022.

[37] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Exp. Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120254. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742300756X

[38] Ü. Öztürk, M. Akdag, and T. Ayabakan, "A review of path planning algorithms in maritime autonomous surface ships: Navigation safety perspective," *Ocean Eng.*, vol. 251, May 2022, Art. no. 111010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801822004334

[39] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of autonomous path planning algorithms for mobile robots," *Drones*, vol. 7, no. 3, p. 211, 2023. [Online]. Available: https://www.mdpi.com/2504-446X/7/3/211

[40] S. Lin, A. Liu, J. Wang, and X. Kong, "A review of path-planning approaches for multiple mobile robots," *Machines*, vol. 10, no. 9, p. 773, Sep. 2022. [Online]. Available: https://www.mdpi.com/2075-1702/10/9/773

[41] M. Jones, S. Djahel, and K. Welsh, "Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey," *ACM Comput. Surveys*, vol. 55, no. 11, pp. 1–39, Nov. 2023.

[42] A. Basiri, V. Mariani, G. Silano, M. Aatif, L. Iannelli, and L. Glielmo, "A survey on the application of path-planning algorithms for multi-rotor UAVs in precision agriculture," *J. Navigat.*, vol. 75, no. 2, pp. 364–383, Mar. 2022.

[43] D. Noh, W. Lee, H.-R. Kim, I.-S. Cho, I.-B. Shim, and S. Baek, "Adaptive coverage path planning policy for a cleaning robot with deep reinforcement learning," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2022, pp. 1–6.

[44] Y. Ma, Y. Zhao, Z. Li, H. Bi, J. Wang, R. Malekian, and M. A. Sotelo, "Cciba*: An improved BA* based collaborative coverage path planning method for multiple unmanned surface mapping vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19578–19588, Oct. 2022.

[45] Y. Liu, W. Zhao, H. Liu, Y. Wang, and X. Yue, "Coverage path planning for robotic quality inspection with control on measurement uncertainty," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 5, pp. 3482–3493, Oct. 2022.

[46] Y. Cao, X. Cheng, and J. Mu, "Concentrated coverage path planning algorithm of UAV formation for aerial photography," *IEEE Sensors J.*, vol. 22, no. 11, pp. 11098–11111, Jun. 2022.

[47] J. Xie and J. Chen, "Multiregional coverage path planning for multiple energy constrained UAVs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17366–17381, Oct. 2022.

[48] L. Jiao, Z. Peng, L. Xi, S. Ding, and J. Cui, "Multi-agent coverage path planning via proximity interaction and cooperation," *IEEE Sensors J.*, vol. 22, no. 6, pp. 6196–6207, Mar. 2022.

[49] L. Wang, Z. Wang, M. Liu, Z. Ying, N. Xu, and Q. Meng, "Full coverage path planning methods of harvesting robot with multi-objective constraints," *J. Intell. Robotic Syst.*, vol. 106, no. 1, pp. 1–15, Sep. 2022.

[50] L. Li, D. Shi, S. Jin, S. Yang, C. Zhou, Y. Lian, and H. Liu, "Exact and heuristic multi-robot Dubins coverage path planning for known environments," *Sensors*, vol. 23, no. 5, p. 2560, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/5/2560

[51] Y. Xia, C. Chen, Y. Liu, J. Shi, and Z. Liu, "Two-layer path planning for multi-area coverage by a cooperative ground vehicle and drone system," *Exp. Syst. Appl.*, vol. 217, May 2023, Art. no. 119604. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423001057

[52] R. I. Mukhamediev, K. Yakunin, M. Aubakirov, I. Assanov, Y. Kuchin, A. Symagulov, V. Levashenko, E. Zaitseva, D. Sokolov, and Y. Amirgaliyev, "Coverage path planning optimization of heterogeneous UAVs group for precision agriculture," *IEEE Access*, vol. 11, pp. 5789–5803, 2023.

[53] V. P. Tran, M. A. Garratt, K. Kasmarik, and S. G. Anavatti, "Dynamic frontier-led swarming: Multi-robot repeated coverage in dynamic environments," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 3, pp. 646–661, Mar. 2023.

[54] G. Fevgas, T. Lagkas, V. Argyriou, and P. Sarigiannidis, "Coverage path planning methods focusing on energy efficient and cooperative strategies for unmanned aerial vehicles," *Sensors*, vol. 22, no. 3, p. 1235, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/1235

[55] M. A. V. J. Muthugala, S. M. B. P. Samarakoon, and M. R. Elara, "Toward energy-efficient online complete coverage path planning of a ship hull maintenance robot based on Glasius bio-inspired neural network," *Exp. Syst. Appl.*, vol. 187, Jan. 2022, Art. no. 115940. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742101294X

[56] N. A. Kyriakakis, M. Marinaki, N. Matsatsinis, and Y. Marinakis, "A cumulative unmanned aerial vehicle routing problem approach for humanitarian coverage path planning," *Eur. J. Oper. Res.*, vol. 300, no. 3, pp. 992–1004, Aug. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721007670

[57] M. A. Luna, M. S. Ale Isaac, A. R. Ragab, P. Campoy, P. F. Peña, and M. Molina, "Fast multi-UAV path planning for optimal area coverage in aerial sensing applications," *Sensors*, vol. 22, no. 6, p. 2297, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/6/2297

[58] S. D. Apostolidis, P. C. Kapoutsis, A. C. Kapoutsis, and E. B. Kosmatopoulos, "Cooperative multi-UAV coverage mission planning platform for remote sensing applications," *Auto. Robots*, vol. 46, no. 2, pp. 373–400, Feb. 2022.

[59] S. M. Swadi, A. K. Kadhim, and G. M. Ali, "Design of path planning controller of autonomous wheeled mobile robot based on triple pendulum behaviour," *Int. J. Mech. Eng. Robot. Res.*, vol. 12, no. 1, pp. 23–31, 2023.

[60] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, vol. 149, pp. 270–299, Jan. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419308539

[61] J. Luo, Z. Wang, M. Xia, L. Wu, Y. Tian, and Y. Chen, "Path planning for UAV communication networks: Related technologies, solutions, and opportunities," *ACM Comput. Surveys*, vol. 55, no. 9, pp. 1–37, Sep. 2023.

[62] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[63] S. Karaman and E. Frazzoli, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 5041–5047.

[64] M. S. Branicky, S. Morgan, J. Levine, and M. M. Curtiss, "Sampling-based planning, control and verification of hybrid systems," *IEE Proc. Control Theory Appl.*, vol. 153, no. 5, pp. 575–590, Sep. 2006.

[65] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[66] M. Elbanhawi and M. Simic, "On the performance of sampling-based optimal motion planners," in *Proc. Eur. Model. Symp.*, Nov. 2013, pp. 73–78.

[67] K. I. Tsianos, I. A. Sucan, and L. E. Kavraki, "Sampling-based robot motion planning: Towards realistic applications," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 2–11, Aug. 2007.

[68] M. Xanthidis, J. M. Esposito, I. Rekleitis, and J. M. O'Kane, "Motion planning by sampling in subspaces of progressively increasing dimension," *J. Intell. Robotic Syst.*, vol. 100, nos. 3–4, pp. 777–789, Dec. 2020.

[69] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.

[70] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. Appl. Comput. Mech.*, vol. 40, no. 5, pp. 1048–1066, 1993.

[71] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. Robot. Res.*, vol. 21, no. 3, pp. 233–255, Mar. 2002.

[72] J. Canny, J. Reif, B. Donald, and P. Xavier, "On the complexity of kinodynamic planning," in *Proc. 29th Annu. Symp. Found. Comput. Sci. (SFCS)*. USA: IEEE Computer Society, 1988, pp. 306–316, doi: 10.1109/SFCS.1988.21947.

[73] J. Canny, A. Rege, and J. Reif, "An exact algorithm for kinodynamic planning in the plane," *Discrete Comput. Geometry*, vol. 6, no. 3, pp. 461–484, Sep. 1991.

[74] R. Alami, J. P. Laumond, and T. Siméon, "Two manipulation planning algorithms," in *Proc. Workshop Algorithmic Found. Robot.* Natick, MA, USA: A K Peters, 1995, pp. 109–125.

[75] D. Berenson, S. S. Srinivasa, D. Ferguson, A. Collet, and J. J. Kuffner, "Manipulation planning with workspace goal regions," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 618–624.

[76] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 625–632.

[77] T. Siméon, J.-P. Laumond, J. Cortés, and A. Sahbani, "Manipulation planning with probabilistic roadmaps," *Int. J. Robot. Res.*, vol. 23, nos. 7–8, pp. 729–746, Aug. 2004.

[78] M. Stilman, J.-U. Schamburek, J. Kuffner, and T. Asfour, "Manipulation planning among movable obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, Oct. 2007, pp. 3327–3332.

[79] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Auton. Robots*, vol. 9, no. 3, pp. 247–253, 2000.

[80] H. Choset, "Coverage for robotics—A survey of recent results," *Ann. Math. Artif. Intell.*, vol. 31, nos. 1–4, pp. 113–126, 2001.

[81] J. I. Vasquez-Gomez, M. Marciano-Melchor, L. Valentin, and J. C. Herrera-Lozada, "Coverage path planning for 2D convex regions," *J. Intell. Robotic Syst.*, vol. 97, no. 1, pp. 81–94, Jan. 2020.

[82] D. Zhu, C. Tian, B. Sun, and C. Luo, "Complete coverage path planning of autonomous underwater vehicle based on GBNN algorithm," *J. Intell. Robotic Syst.*, vol. 94, no. 1, pp. 237–249, Apr. 2019.

[83] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, "Energy-constrained multi-UAV coverage path planning for an aerial imagery mission using column generation," *J. Intell. Robotic Syst.*, vol. 97, no. 1, pp. 125–139, Jan. 2020.

[84] P.-M. Hsu, C.-L. Lin, and M.-Y. Yang, "On the complete coverage path planning for mobile robots," *J. Intell. Robotic Syst.*, vol. 74, nos. 3–4, pp. 945–963, Jun. 2014.

[85] C. Di Franco and G. Buttazzo, "Coverage path planning for UAVs photogrammetry with energy and resolution constraints," *J. Intell. Robotic Syst.*, vol. 83, nos. 3–4, pp. 445–462, Sep. 2016.

[86] I. A. Hameed, "Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain," *J. Intell. Robotic Syst.*, vol. 74, nos. 3–4, pp. 965–983, Jun. 2014.

[87] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, "DARP: Divide areas algorithm for optimal multi-robot coverage path planning," *J. Intell. Robotic Syst.*, vol. 86, nos. 3–4, pp. 663–680, Jun. 2017.

[88] Y. Li, D. Li, C. Maple, Y. Yue, and J. Oyekan, "K-order surrounding roadmaps path planner for robot path planning," *J. Intell. Robotic Syst.*, vol. 75, nos. 3–4, pp. 493–516, Sep. 2014.

[89] J. Faigl, M. Kulich, and L. Přeučil, "A sensor placement algorithm for a mobile robot inspection planning," *J. Intell. Robotic Syst.*, vol. 62, nos. 3–4, pp. 329–353, Jun. 2011.

[90] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1258–1276, Dec. 2013.

[91] T. Oksanen and A. Visala, "Coverage path planning algorithms for agricultural field machines," *J. Field Robot.*, vol. 26, no. 8, pp. 651–668, Aug. 2009.

[92] A. Khan, I. Noreen, and Z. Habib, "On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges," *J. Inf. Sci. Eng.*, vol. 33, no. 1, pp. 101–121, 2017.

[93] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA, USA: MIT Press, 2005.

[94] A. Zelinsky, R. Jarvis, J. C. Byrne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proc. Int. Conf. Adv. Robot.*, 1993, pp. 533–538.

[95] P. N. Atkar, D. C. Conner, A. Greenfield, H. Choset, and A. A. Rizzi, "Hierarchical segmentation of piecewise pseudoextruded surfaces for uniform coverage," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 107–120, Jan. 2009.

[96] E. Galceran and M. Carreras, "Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4159–4164.

[97] J. Jin and L. Tang, "Coverage path planning on three-dimensional terrain for arable farming," *J. Field Robot.*, vol. 28, no. 3, pp. 424–440, May 2011.

[98] G. Papadopoulos, H. Kurniawati, and N. M. Patrikalakis, "Asymptotically optimal inspection planning using systems with differential constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4126–4133.

[99] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proc. ICRA IEEE Int. Conf. Robot. Autom.*, vol. 1, 2001, pp. 27–32.

[100] P. A. Jimenez, B. Shirinzadeh, A. Nicholson, and G. Alici, "Optimal area covering using genetic algorithms," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. mechatronics*, Sep. 2007, pp. 1–5.

[101] T. Bretl and S. Hutchinson, "Robust coverage by a mobile robot of a planar workspace," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4582–4587.

[102] E. Galceran, S. Nagappa, M. Carreras, P. Ridao, and A. Palomer, "Uncertainty-driven survey path planning for bathymetric mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 6006–6012.

[103] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3852–3857.

[104] K. Easton and J. Burdick, "A coverage algorithm for multi-robot boundary inspection," in *Proc. IEEE Int. Conf. Robot. Autom.*, Oct. 2005, pp. 727–734.

[105] A. Ahmadzadeh, J. Keller, G. Pappas, A. Jadbabaie, and V. Kumar, *An Optimization-Based Approach to Time-Critical Cooperative Surveillance and Coverage With UAVs*. Berlin, Germany: Springer, 2008, pp. 491–500, doi: 10.1007/978-3-540-77457-0_46.

[106] E. I. Grøtli and T. A. Johansen, "Path planning for UAVs under communication constraints using SPLAT! And MILP," *J. Intell. Robotic Syst.*, vol. 65, nos. 1–4, pp. 265–282, Jan. 2012.

[107] Y.-H. Choi, T.-K. Lee, S.-H. Baek, and S.-Y. Oh, "Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 5788–5793.

[108] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor-driven online coverage planning for autonomous underwater vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 6, pp. 1827–1838, Dec. 2013.

[109] Z. J. Butler, A. A. Rizzi, and R. L. Hollis, "Contact sensor-based coverage of rectilinear environments," in *Proc. IEEE Int. Symp. Intell. Control Intell. Syst. Semiotics*, Sep. 1999, pp. 266–271.

[110] Y. Gabriely and E. Rimon, "Spiral-STC: An on-line coverage algorithm of grid environments by a mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 2002, pp. 954–960.

[111] C. Luo, S. Yang, D. Stacey, and J. Jofriet, "A solution to vicinity problem of obstacles in complete coverage path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Jul. 2002, pp. 612–617.

[112] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Syst., Man Cybern., B Cybern.*, vol. 34, no. 1, pp. 718–724, Feb. 2004.

[113] S. Hert, S. Tiwari, and V. Lumelsky, *A Terrain Covering Algorithm for an AUV*. Cham, Switzerland: Springer, 1996, pp. 17–45.

[114] P. N. Atkar, H. Choset, A. A. Rizzi, and E. U. Acar, "Exact cellular decomposition of closed orientable surfaces embedded in /spl Rfr//sup 3," in *Proc. ICRA IEEE Int. Conf. Robot. Autom.*, vol. 1, 2001, pp. 699–704.

[115] P. Cheng, J. Keller, and V. Kumar, "Time-optimal UAV trajectory planning for 3D urban structure coverage," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2008, pp. 2750–2757.

[116] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 5525–5530.

[117] E. U. Acar and H. Choset, "Exploiting critical points to reduce positioning error for sensor-based navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, 2002, pp. 3831–3837.

[118] A. Kim, "Active visual SLAM with exploration for autonomous underwater navigation," thesis, Dept. Mech. Eng., Michigan Univ. Ann Arbor, Ann Arbor, MI, USA, 2012.

[119] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: An algorithmic approach," *Ann. Math. Artif. Intell.*, vol. 52, nos. 2–4, pp. 109–142, Apr. 2008.

[120] S. Choi, S. Lee, H. H. Viet, and T. Chung, "B-Theta*: An efficient online coverage algorithm for autonomous cleaning robots," *J. Intell. Robotic Syst.*, vol. 87, no. 2, pp. 265–290, Aug. 2017.

[121] Y. Nakamura and A. Sekiguchi, "The chaotic mobile robot," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 898–904, Oct. 2001.

[122] R. Gilmore and M. Lefranc, *The Topology of Chaos*. College Park, MD, USA: AAPT, 2003.

[123] J. Awrejcewicz, *Bifurcation and Chaos: Theory and Applications*. Berlin, Germany: Springer, 2012.

[124] X. Zang, S. Iqbal, Y. Zhu, X. Liu, and J. Zhao, "Applications of chaotic dynamics in robotics," *Int. J. Adv. Robotic Syst.*, vol. 13, no. 2, p. 60, Mar. 2016.

[125] V. Shivashankar, R. Jain, U. Kuter, and D. Nau, "Real-time planning for covering an initially-unknown spatial environment," in *Proc. FLAIRS Conf.*, 2011, pp. 63–68.

[126] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra, and C. Bustacara, "BSA: A complete coverage algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 2040–2044.

[127] T.-K. Lee, S.-H. Baek, Y.-H. Choi, and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robot. Auto. Syst.*, vol. 59, no. 10, pp. 801–812, Oct. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889011000996

[128] C. Luo and S. X. Yang, "A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1279–1298, Jul. 2008.

[129] M. Z. Yan and D. Q. Zhu, "An algorithm of complete coverage path planning for autonomous underwater vehicles," in *Materials, Mechatronics and Automation* (Key Engineering Materials), vol. 467. Stafa-Zurich, Switzerland: Trans Tech Publications, May 2011, pp. 1377–1385.

[130] X. Qiu, J. Song, X. Zhang, and S. Liu, "A complete coverage path planning method for mobile robot in uncertain environments," in *Proc. 6th World Congr. Intell. Control Autom.*, vol. 2, 2006, pp. 8892–8896.

[131] Y. Guo and M. Balakrishnan, "Complete coverage control for nonholonomic mobile robots in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jul. 2006, pp. 1704–1709.

[132] J. Palacin, T. Palleja, I. Valganon, R. Pernia, and J. Roca, "Measuring coverage performances of a floor cleaning mobile robot using a vision system," in *Proc. IEEE Int. Conf. Robot. Autom.*, Oct. 2005, pp. 4236–4241.

[133] T.-S. Lee, J.-S. Choi, J.-H. Lee, and B.-H. Lee, "3-D terrain covering and map building algorithm for an AUV," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 4420–4425.

[134] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and Service Robotics*, A. Zelinsky, Ed. London, U.K.: Springer, 1998, pp. 203–209.

[135] H. Choset, E. Acar, A. Rizzi, and J. Luntz, "Exact cellular decompositions in terms of critical points of Morse functions," in *Proc. ICRA Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia*, vol. 3, 2000, pp. 2270–2277.

[136] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 331–344, Apr. 2002.

[137] J. Milnor, *Morse Theory (AM-51)*, vol. 51. Princeton, NJ, USA: Princeton Univ. Press, 2016.

[138] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of Morse decompositions," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 345–366, Apr. 2002.

[139] E. U. Acar and H. Choset, "Robust sensor-based coverage of unstructured environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Expanding Societal Role Robot. the Next Millennium*, vol. 1, 2001, pp. 61–68.

[140] E. A. Codling, M. J. Plank, and S. Benhamou, "Random walk models in biology," *J. Roy. Soc. Interface*, vol. 5, no. 25, pp. 813–834, 2008.

[141] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, "A swarm robotic exploration strategy based on an improved random walk method," *J. Robot.*, vol. 2019, pp. 1–9, Mar. 2019.

[142] B. Pang, J. Qi, C. Zhang, Y. Song, and R. Yang, "Analysis of random walk models in swarm robots for area exploration," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 2484–2489.

[143] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. da Luz, E. P. Raposo, and H. E. Stanley, "Optimizing the success of random searches," *Nature*, vol. 401, no. 6756, pp. 911–914, Oct. 1999.

[144] D. W. Sims, N. E. Humphries, R. W. Bradford, and B. D. Bruce, "Lévy flight and Brownian search patterns of a free-ranging predator reflect different prey field characteristics," *J. Animal Ecology*, vol. 81, no. 2, pp. 432–442, Mar. 2012.

[145] T. Balch, "The case for randomized search," in *Proc. Workshop Sensors Motion, IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, USA, Apr. 2000, pp. 213–215.

[146] D. W. Gage, "Randomized search strategies with imperfect sensors," in *Proc. SPIE*, W. J. Wolfe and W. H. Chun, Eds. vol. 2058, 1994, pp. 270–279, doi: 10.1117/12.167503.

[147] R. Fujisawa and S. Dobata, "Lévy walk enhances efficiency of group foraging in pheromone-communicating swarm robots," in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, Dec. 2013, pp. 808–813.

[148] A. Schroeder, S. Ramakrishnan, M. Kumar, and B. Trease, "Efficient spatial coverage by a robot swarm based on an ant foraging model and the Lévy distribution," *Swarm Intell.*, vol. 11, no. 1, pp. 39–69, Mar. 2017.

[149] D. K. Sutantyo, S. Kernbach, P. Levi, and V. A. Nepomnyashchikh, "Multi-robot searching algorithm using Lévy flight and artificial potential field," in *Proc. IEEE Saf. Secur. Rescue Robot.*, 2010, pp. 1–6.

[150] A. Deshpande, M. Kumar, and S. Ramakrishnan, "Robot swarm for efficient area coverage inspired by ant foraging: The case of adaptive switching between Brownian motion and Lévy flight," in *Proc. Mechtron., Estimation Identificat.; Uncertain Syst. Robustness; Path Planning Motion Control; Tracking Control Syst.; Multi-Agent Networked Syst.; Manuf.; Intell. Transp. Vehicles; Sensors Actuators; Diagnostics (Don't short) Detection; Unmanned, Ground Surface Robotics; Motion Vib. Control Appl.*, vol. 2. New York, NY, USA: American Society of Mechanical Engineers, Oct. 2017, p. 8, doi: 10.1115/dscc2017-5229.

[151] H. Steven, *Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, and Engineering*. Boulder, CO, USA: Westview Press, 1994.

[152] D.-I. Curiac and C. Volosencu, "Path planning algorithm based on Arnold cat map for surveillance UAVs," *Defence Sci. J.*, vol. 65, no. 6, pp. 483–488, Nov. 2015.

[153] V. S. Afraimovich and S.-B. Hsu, *Lectures on Chaotic Dynamical Systems*. Providence, RI, USA: American Mathematical Society, 2003.

[154] R. Femat and G. Solis-Perales, *Robust Synchronization of Chaotic Systems via Feedback*, vol. 378. Cham, Switzerland: Springer, 2009.

[155] G. Chen and T. Ueta, "Yet another chaotic attractor," *Int. J. Bifurcation Chaos*, vol. 9, no. 7, pp. 1465–1466, Jul. 1999.

[156] J. Lü and G. Chen, "A new chaotic attractor coined," *Int. J. Bifurcation Chaos*, vol. 12, no. 3, pp. 659–661, Mar. 2002.

[157] S.-M. Guo, L. S. Shieh, G. Chen, and M. Ortega, "Ordering chaos in chua's circuit: A sampled-data feedback and digital redesign approach," *Int. J. Bifurcation Chaos*, vol. 10, no. 9, pp. 2221–2231, Sep. 2000.

[158] G. Chen and X. Dong, "Ordering chaos of Chua's circuit—A feedback control approach," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1993, pp. 2604–2607.

[159] C. Nwachioma and J. H. Pérez-Cruz, "Analysis of a new chaotic system, electronic realization and use in navigation of differential drive mobile robot," *Chaos, Solitons Fractals*, vol. 144, Mar. 2021, Art. no. 110684. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077921000370

[160] K. Sridharan, P. McNamee, Z. N. Ahmadabadi, and J. Hudack, "Online search of unknown terrains using a dynamical system-based path planning approach," 2021, *arXiv:2103.11863*.

[161] U. P. Amadasun, P. McNamee, and Z. N. Ahmadabadi, "Autonomous search of real-life environments combining dynamical system-based path planning and unsupervised learning," 2023, *arXiv:2305.01834*.

[162] K. Sridharan and Z. N. Ahmadabadi, "A multi-system chaotic path planner for fast and unpredictable online coverage of terrains," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5268–5275, Oct. 2020.

[163] C. Li, Y. Song, F. Wang, Z. Wang, and Y. Li, "A bounded strategy of the mobile robot coverage path planning based on Lorenz chaotic system," *Int. J. Adv. Robotic Syst.*, vol. 13, no. 3, p. 107, May 2016.

[164] M. W. Hirsch, *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, 3rd ed. Waltham, MA, USA: Academic Press, 2013.

[165] D.-I. Curiac and C. Volosencu, "Chaotic trajectory design for monitoring an arbitrary number of specified locations using points of interest," *Math. Problems Eng.*, vol. 2012, pp. 1–18, 2012.

[166] C. Li, C. Liu, Y. Song, and Z. Liang, "Parameter value selection strategy for complete coverage path planning based on the lü system to perform specific types of missions," *Frontiers Inf. Technol. Electron. Eng.*, vol. 24, no. 2, pp. 231–244, Feb. 2023.

[167] L. Chua, "The genesis of Chua's circuit," *Archiv für Elektronik und Übertragungstechnik*, vol. 46, no. 4, pp. 250–257, 1992.

[168] C. K. Volos, N. G. Bardis, I. M. Kyprianidis, and I. N. Stouboulos, "Implementation of mobile robot by using double-scroll chaotic attractors," in *Proc. 11th Int. Conf. Appl. Elect. Comput. Eng. (ACA)*, Athens, Greece. Stevens Point, WI, USA: World Scientific and Engineering Academy and Society (WSEAS), 2012, pp. 119–124.

[169] C. Pimentel-Romero, J. M. Muñoz-Pacheco, O. Felix-Beltran, L. Gomez-Pavon, and C. K. Volos, *Chaotic Planning Paths Generators by Using Performance Surfaces*. Cham, Switzerland: Springer, 2017, pp. 805–832.

[170] S. Nasr, H. Mekki, and K. Bouallegue, "A multi-scroll chaotic system for a higher coverage path planning of a mobile robot using flatness controller," *Chaos, Solitons Fractals*, vol. 118, pp. 366–375, Jan. 2019.

[171] E. Tlelo-Cuautle, H. C. Ramos-López, M. Sánchez-Sánchez, A. D. Pano-Azucena, L. A. Sánchez-Gaspariano, J. C. Núñez-Pérez, and J. L. Camas-Anzueto, "Application of a chaotic oscillator in an autonomous mobile robot," *J. Electr. Eng.*, vol. 65, no. 3, pp. 157–162, May 2014.

[172] C. K. Volos, I. M. Kyprianidis, and I. N. Stouboulos, "A chaotic path planning generator for autonomous mobile robots," *Robot. Auto. Syst.*, vol. 60, no. 4, pp. 651–656, Apr. 2012.

[173] L. Moysis, E. Petavratzis, M. Marwan, C. Volos, H. Nistazakis, and S. Ahmad, "Analysis, synchronization, and robotic application of a modified hyperjerk chaotic system," *Complexity*, vol. 2020, pp. 1–15, Apr. 2020.

[174] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, Jun. 1976.

[175] C. Li, F. Wang, L. Zhao, Y. Li, and Y. Song, "An improved chaotic motion path planner for autonomous mobile robots based on a logistic map," *Int. J. Adv. Robotic Syst.*, vol. 10, no. 6, p. 273, Jun. 2013.

[176] E. Petavratzis, L. Moysis, C. Volos, M. K. Gupta, I. Stouboulos, and S. Goudos, "Chaotic motion control of a mobile robot using a memory technique," in *Proc. 24th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2020, pp. 506–511.

[177] E. Petavratzis, L. Moysis, C. Volos, H. Nistazakis, and I. Stouboulos, "Chaotic path planning for grid coverage using a modified logistic-may map," *J. Autom., Mobile Robot. Intell. Syst.*, vol. 14, no. 2, pp. 3–9, 2020.

[178] L. Moysis, E. Petavratzis, C. Volos, H. Nistazakis, I. Stouboulos, and K. Valavanis, "A chaotic path planning method for 3D area coverage using modified logistic map and a modulo tactic," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 220–227.

[179] L. Moysis, E. Petavratzis, C. Volos, H. Nistazakis, and I. Stouboulos, "A chaotic path planning generator based on logistic map and modulo tactics," *Robot. Auto. Syst.*, vol. 124, Feb. 2020, Art. no. 103377.

[180] E. Petavratzis, L. Moysis, C. Volos, I. Stouboulos, H. Nistazakis, and K. Valavanis, "A chaotic path planning generator enhanced by a memory technique," *Robot. Auto. Syst.*, vol. 143, Sep. 2021, Art. no. 103826. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889021001111

[181] D.-I. Curiac and C. Volosencu, "A 2D chaotic path planning for mobile robots accomplishing boundary surveillance missions in adversarial conditions," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 10, pp. 3617–3627, Oct. 2014.

[182] S. Hongyan, C. Zhifei, and S. Changzhi, "Application of improved chaotic optimization algorithm to problems with motion planning for mobile robots," in *Proc. IEEE Int. Conf. Robot., Intell. Syst. Signal Process.*, Jul. 2003, pp. 611–615.

[183] E. Petavratzis, C. Volos, A. Ouannas, H. Nistazakis, K. Valavanis, and I. Stouboulos, "A 2D discrete chaotic memristive map and its application in robot's path planning," in *Proc. 10th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, Jul. 2021, pp. 1–4.

[184] L. Moysis, K. Rajagopal, A. V. Tutueva, C. Volos, B. Teka, and D. N. Butusov, "Chaotic path planning for 3D area coverage using a pseudo-random bit generator from a 1D chaotic map," *Mathematics*, vol. 9, no. 15, p. 1821, 2021. [Online]. Available: https://www.mdpi.com/2227-7390/9/15/1821

[185] E. K. Petavratzis, C. K. Volos, I. N. Stouboulos, H. E. Nistazakis, K. G. Kyritsi, and K. P. Valavanis, "Coverage performance of a chaotic mobile robot using an inverse pheromone model," in *Proc. 8th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, May 2019, pp. 1–4.

[186] E. Petavratzis, C. Volos, L. Moysis, H. Nistazakis, A. Giakoumis, and I. Stouboulos, "Experimental coverage performance of a chaotic autonomous mobile robot," in *Proc. 11th Int. Conf. Modern Circuits Syst. Technol. (MOCAST)*, Jun. 2022, pp. 1–4.

[187] E. Petavratzis, C. Volos, and I. Stouboulos, "Experimental study of terrain coverage of an autonomous chaotic mobile robot," *Integration*, vol. 90, pp. 104–114, May 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167926023000160

[188] A. Jansri, K. Klonikarn, and P. Sooraksa, "On comparison of attractors for chaotic mobile robots," in *Proc. 30th Annu. Conf. IEEE Ind. Electron. Soc.*, vol. 3, 2004, pp. 2536–2541.

[189] A. J. Kitdakom Klomkarn and P. Sooraksa, "Further investigation on trajectory of chaotic guiding signals for robotic systems," in *Proc. IEEE Int. Symp. Commun. Inf. Technol.*, vol. 2, 2004, pp. 1166–1170.

[190] C. K. Volos, N. Bardis, I. M. Kyprianidis, and I. N. Stouboulos, "Motion control of a mobile robot based on double-scroll chaotic circuits," *WSEAS Trans. Syst.*, vol. 11, no. 9, pp. 479–488, 2012.

[191] C. K. Volos, I. M. Kyprianidis, I. N. Stouboulos, and A. N. Anagnostopoulos, "Experimental study of the dynamic behavior of a double scroll circuit," *J. Appl. Funct. Anal.*, vol. 4, no. 4, pp. 703–711, 2009.

[192] J. J. Cetina-Denis, R. M. Lopéz-Gutiérrez, C. Cruz-Hernández, and A. Arellano-Delgado, "Design of a chaotic trajectory generator algorithm for mobile robots," *Appl. Sci.*, vol. 12, no. 5, p. 2587, Mar. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/5/2587

[193] A. Ibrahim Majeed, "Mobile robot motion control based on chaotic trajectory generation," *J. Eng. Sustain. Develop.*, vol. 24, no. 4, pp. 48–55, Jul. 2020. [Online]. Available: https://jeasd.uomustansiriyah.edu.iq/index.

[194] C.-H. Li, Y. Song, F.-Y. Wang, Z.-Q. Wang, and Y.-B. Li, "A chaotic coverage path planner for the mobile robot based on the Chebyshev map for special missions," *Frontiers Inf. Technol. Electron. Eng.*, vol. 18, no. 9, pp. 1305–1319, Sep. 2017.

[195] K. Fallahi and H. Leung, "A cooperative mobile robot task assignment and coverage planning based on chaos synchronization," *Int. J. Bifurcation Chaos*, vol. 20, no. 1, pp. 161–176, Jan. 2010.

[196] A. Zhu and H. Leung, "Cooperation random mobile robots based on chaos synchronization," in *Proc. IEEE Mechatronics Int. Conf.*, May 2007, pp. 1–5.

[197] C. K. Volos, I. M. Kyprianidis, and I. N. Stouboulos, "Experimental investigation on coverage performance of a chaotic autonomous mobile robot," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1314–1322, Dec. 2013.

[198] P. Sooraska and K. Klomkarn, "'No-CPU' chaotic robots: From classroom to commerce," *IEEE Circuits Syst. Mag.*, vol. 10, no. 1, pp. 46–53, Oct. 2010.

[199] Y. Bae, "Obstacle avoidance method in the chaotic robot," in *Proc. 23rd Digit. Avionics Syst. Conf.*, 2004, p. 12.

[200] C. Li, Y. Song, F. Wang, Z. Liang, and B. Zhu, "Chaotic path planner of autonomous mobile robots based on the standard map for surveillance missions," *Math. Problems Eng.*, vol. 2015, pp. 1–11, 2015.

[201] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.*, vol. 64, no. 8, p. 821, Feb. 1990.

[202] J.-H. Lin and L.-R. Huang, "Chaotic bee swarm optimization algorithm for path planning of mobile robots," in *Proc. 10th WSEAS Int. Conf. Evol. Comput.* Stevens Point, WI, USA: World Scientific and Engineering Academy and Society (WSEAS), 2009, pp. 84–89.

[203] L. M. Pecora, T. L. Carroll, G. A. Johnson, D. J. Mar, and J. F. Heagy, "Fundamentals of synchronization in chaotic systems, concepts, and applications," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 7, no. 4, pp. 520–543, Dec. 1997.

[204] M. Ma, J. Wu, Y. Shi, L. Yue, C. Yang, and X. Chen, "Chaotic random opposition-based learning and Cauchy mutation improved moth-flame optimization algorithm for intelligent route planning of multiple UAVs," *IEEE Access*, vol. 10, pp. 49385–49397, 2022.

[205] C. Choi, S.-G. Hong, J.-H. Shin, I.-K. Jeong, and J.-J. Lee, "Dynamical path-planning algorithm of a mobile robot using chaotic neuron model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Hum. Robot Interact. Cooperat. Robots*, vol. 2, 1995, pp. 456–461.

[206] C. Zhang, "Path planning for robot based on chaotic artificial potential field method," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 317, Mar. 2018, Art. no. 012056, doi: 10.1088/1757-899x/317/1/012056.

[207] J. Lian, W. Yu, K. Xiao, and W. Liu, "Cubic spline interpolation-based robot path planning using a chaotic adaptive particle swarm optimization algorithm," *Math. Problems Eng.*, vol. 2020, pp. 1–20, Feb. 2020.

[208] A. Tharwat, M. Elhoseny, A. E. Hassanien, T. Gabel, and A. Kumar, "Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm," *Cluster Comput.*, vol. 22, no. 4, pp. 4745–4766, Mar. 2018, doi: 10.1007/s10586-018-2360-3.

[209] Z. Wang and J. Cai, "The path-planning in radioactive environment of nuclear facilities using an improved particle swarm optimization algorithm," *Nucl. Eng. Design*, vol. 326, pp. 79–86, Jan. 2018.

[210] H. Chu, J. Yi, and F. Yang, "Chaos particle swarm optimization enhancement algorithm for UAV safe path planning," *Appl. Sci.*, vol. 12, no. 18, p. 8977, Sep. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/18/8977

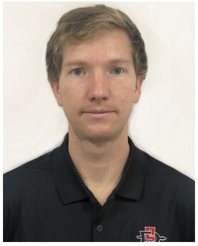
[211] L. Cao and Q. Chen, "Cooperative path planning for multi-USV based on improved artificial bee colony algorithm," in *Proc. SPIE*, N. Sang, J. Ma, and Z. Chen, Eds. vol. 10611, 2018, pp. 352–359, doi: 10.1117/12.2285623.

[212] D.-I. Curiac, O. Banias, C. Volosencu, and C.-D. Curiac, "Novel bioinspired approach based on chaotic dynamics for robot patrolling missions with adversaries," *Entropy*, vol. 20, no. 5, p. 378, May 2018.

[213] Y. Bae, "Target searching method in the chaotic mobile robot," in *Proc. 23rd Digit. Avionics Syst. Conf.*, 2004, p. 12.

[214] R. Méndez-Ramírez, A. Arellano-Delgado, M. Murillo-Escobar, and C. Cruz-Hernández, "Degradation analysis of chaotic systems and their digital implementation in embedded systems," *Complexity*, vol. 2019, Dec. 2019, Art. no. 9863982, doi: 10.1155/2019/9863982.

FARIDA AHURAKA, photograph and biography not available at the time of publication.



PATRICK MCNAMEE received the B.S. and M.Sc. degrees in aerospace engineering and the M.Sc. degree in computer science from the University of Kansas, Lawrence, KS, USA, in 2017, 2020, and 2021, respectively.

From 2016 to 2021, he was an Undergraduate and Graduate Student Researcher with the University of Kansas, where he studied the dynamics and controls of unmanned aerial vehicles (UAVs) and machine learning applications for UAVs. He has been a Graduate Student Researcher with the Dynamic Systems and Intelligent Machines Laboratory (DSIM), San Diego State University, since 2021.

QIXU WANG received the B.Sc. degree in process equipment and control engineering from the Dalian University of Technology, Dalian, Liaoning, China, in 2019, and the M.Sc. degree in mechanical engineering from Northeastern University, Boston, MA, USA, in 2021. He has been a Graduate Student Researcher with the Dynamic Systems and Intelligent Machines Laboratory (DSIM), San Diego State University, since 2022.

ZAHRA NILI AHMADABADI (Member, IEEE) is currently an Assistant Professor with the Mechanical Engineering Department and the Director of the Dynamic Systems and Intelligent Machines Laboratory, San Diego State University (SDSU). Prior to this position, she was an Assistant Professor with Wichita State University. Her current research interests include acoustic perception, deep learning and sequence modeling, cooperative robotics, and nonlinear dynamical systems.

JEFFREY HUDACK (Member, IEEE) is currently a Senior Computer Scientist with the Air Force Research Laboratory (AFRL), specializing in distributed, multi-agent systems, and machine learning. He leads research on self-organizing systems that can act autonomously while maintaining coordinated global behavior, with applications for swarms of air vehicles, federated learning systems, and distributed command and control. He has been recognized as an Outstanding Mentor by the AFRL Scholars Program, with research collaborations on emerging applications of artificial intelligence to distributed systems.