Experimental Coverage Performance of a Chaotic Autonomous Mobile Robot

Eleftherios Petavratzis
Laboratory of Nonlinear Systems - Circuits & Complexity
Department of Physics,
Aristotle University of Thessaloniki
Thessaloniki, Greece
elpetavr@physics.auth.gr

Christos Volos
Laboratory of Nonlinear Systems - Circuits & Complexity
Department of Physics,
Aristotle University of Thessaloniki
Thessaloniki, Greece
volos@physics.auth.gr

Lazaros Moysis
Laboratory of Nonlinear Systems - Circuits & Complexity
Department of Physics,
Aristotle University of Thessaloniki
Thessaloniki, Greece
lmousis@physics.auth.gr

Hector Nistazakis
Section of Electronic Physics and Systems,
Department of Physics,
National and Kapodistrian University of Athens,
Athens, Greece
enistaz@phys.uoa.gr

Angelos Giakoumis
Department of Information and Electronic Engineering,
International Hellenic University
Thessaloniki, Greece
ang1960@ihu.gr

Ioannis Stouboulos
Laboratory of Nonlinear Systems - Circuits & Complexity
Department of Physics,
Aristotle University of Thessaloniki
Thessaloniki, Greece
stouboulos@physics.auth.gr

***Abstract*—This work, presents the design and the implementation of a mobile robot that can cover given workspaces. The motion commands have been produced by a chaotic system which gives high "randomness" to its behavior. For the implementation of the robot a microcontroller has been used (Arduino Uno). Moreover, the obstacle avoidance has been achieved by using an ultrasonic sensor, which enables the tracking and avoidance of placed obstacles. As a result, the motion trajectory becomes smoother. The robot's behavior has been studied in scenarios with different starting positions showing promising results.**

***Index Terms*—Chaos, Logistic Map, Path Planning, Terrain Coverage, Microcontroller, Area Coverage**

I. Introduction

The production of powerful computers triggered an explosion in technological development and led to the construction of autonomous robotic systems that were able to complete extreme complex missions. In the industrial area, automated mass production is considered to be the starting point for the use of autonomous robotic systems. The need for mass production in industry was covered by the use of robotic systems. Soon, the use of robots was integrated into domestic applications (see for example vacuum cleaners) [1], [2].

Especially, robotic systems like these are based on the study of path planning. Path planning is actually a constructed strategy, in which the basic aim is the motion from a starting point A to an ending point B. Also, this behavior can be used for the exploration of a given area [3], [4]. The strategies that have been proposed, managed to create trajectories that could cover efficiently the area of interest [5], [6]. The implementation of these techniques was another challenging task. In order to construct these applications, microcontrollers have been used (Arduino/Raspberry Pi) [7].

Also, for these applications, chaotic systems have been widely used. This technique which is known as chaotic area coverage, enables the ability of producing a "random" motion by a robot into a given workspace. Their simplicity and behavior that is easy to understand, made the chaotic systems so popular among scientists [8], [9]. Furthermore, their uniqueness lies on the fact that chaotic systems are sensitive to changes in initial conditions and parameters. That means, that even a minor change could force them to produce unpredictable results.

In this work, the logistic chaotic map is selected for producing a Chaotic Pseudo Random Bit Generator (CPRBG). In more details, the values of the chaotic map can be used to generate bits with the use of a simple rule. Then, the necessary motion commands in four directions are generated based on the produced bitstream. According to the proposed motion strategy the workspace is divided into cells of dimensions $32.5cm \times 32.5cm$. In order to avoid the workspace's boundaries, the robot can detect if its next motion command lead it to an area with an obstacle. If this unacceptable behavior is observed, the robot is searching if there are unblockable paths next to placed obstacle. If there is only one path, then it follows that path. If there are two available paths then according to a modulo operator the robot selects in which path will move in. On the other hand, if there are no available path then it stops its motion, moves for a small amount of time backwards and waits for the next motion command. With this technique it manages to use the majority of the produced motion commands.

The next task is to implement the produced technique. The implementation is held into an Arduino Uno microcontroller which is placed in a AlphaBot platform [10]. The robot has three wheels and one ultrasonic sensor (HC-SR04), which can detect the distance of each obstacle. With the use of the ultrasonic sensor the robot avoids placed obstacles and visits unexplored areas.

The structure of the paper is organized as follows. In Section II, the proposed pseudo random bit generator, which is based on the logistic map, is presented. In Section III the proposed pseudo random bit generator, which is implemented into an Arduino microcontroller, is described. In Section IV, there is the conclusion of the this work and some thoughts of future extensions.

II. The proposed CRBG

A. The Logistic Map

The system that has been chosen for producing the motion commands is the typical logistic map which is presented by the iterative map:

$$x_{k+1} = r \cdot x_k \cdot (1 - x_k), \quad 0 \leq x \leq 1, \tag{1}$$

where the parameter *r* varies in the interval $[0, 4]$. The bifurcation diagram of the logistic map is shown in Fig. 1. From this diagram a route to chaos through a period-doubling sequence is observed. In more details, for $3.5 \leq r \leq 4$, the system goes from a periodic trajectory into chaos with a small windows of periodic behavior. In this work the value of $r = 4$ is selected, while the initial condition is $x_0 = 0.1$.

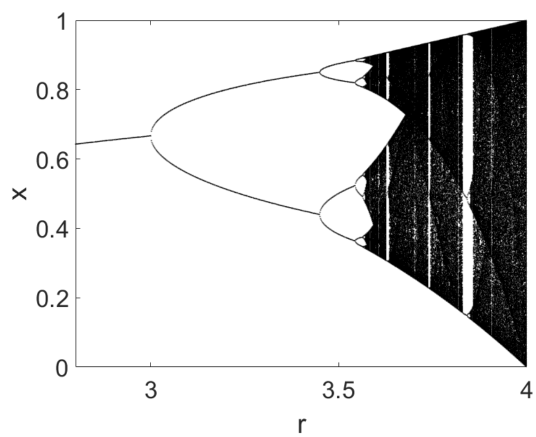

Fig. 1. The bifuraction diagram of the logistc map.

B. Chaotic Random Bit Generator

The next task is to produce the necessary bitstreams. For that purpose, the produced values of logistic map of Eq. (1) are used to generate bits using the following rule:

$$b_i = \begin{cases} [10^4 \cdot x_i] \pmod 2, & 0 \leq x_i < 0.1 \\ 0, & 0.1 \leq x_i < 0.5 \\ 1, & 0.5 \leq x_i < 0.9 \\ [10^4 \cdot x_i] \pmod 2, & 0.9 \leq x_i \leq 1 \end{cases} \tag{2}$$

where $[\cdot]$ denotes the integer part of the argument. Next, the validation of the randomness of the produced bitstream is essential. In order to do that the bitstream should be checked by FIPS 140-2 Statistical Test Suite [11] if it can pass the statistical tests. These tests were applied for $m = 100$ groups of sequences of $1,000,000$ bits. The significance level was selected as $\alpha = 0.01$ and according to the results, which are shown in Table I, the bitstream managed to pass all the necessary test confirming the claim that the bit sequence is random.

TABLE I
FIPS 140-2 STATISTICAL TEST RESULTS

If $P \geq \alpha$ then the test is successful			
No.	**Test**	**P-Value**	**Status**
1	Frequency	0.616305	success
2	BlockFrequency	0.834308	success
3	CumulativeSums	0.334538	success
4	Runs	0.574903	success
5	LongestRun	0.964295	success
6	Rank	0.6786864	success
7	FFT	0.5141246	success
8	NonOverlappingTemplate	0.994250	success
9	OverlappingTemplate	0.474986	success
10	Universal	0.911413	success
11	ApproximateEntropy	0.350485	success
12	RandomExcursions	0.862344	success
13	RandomExcursionsVariant	0.911413	success
14	Serial	0.946308	success
15	LinearComplexity	0.924076	success

C. Motion in Four Directions

Next the robot uses the values of the bitstream for its motion. The robot moves into four different directions with the use of four motions, as it is shown in Table II. In each iteration a pair of bits is considered for giving the motion command. In order to avoid collision with the boundaries of the workspace, the robot can turn 45 degrees either clockwise or counter clockwise. If both options are acceptable then the values of logistic map (1) are used in Eq.(3).

$$c_i = \text{mod}([10^4 \cdot (x_i)], 2) \tag{3}$$

Therefore, according to values of c_i we have two cases.

$$c_i = \begin{cases} 0, & \text{the robot turns 45 degrees counter clockwise} \\ 1, & \text{the robot turns 45 degrees clockwise} \end{cases} \tag{4}$$

When both cases are forbidden due to the presence of an obstacle then the robot moves backwards and waits for another motion command.

TABLE II
BIT SEQUENCE AND MOTION COMMANDS

Motion in 4 directions	
Bits	**Motion command**
00	forward
01	90 degrees turn (clockwise)
10	180 degrees turn (clockwise)
11	90 degress turn (counter clockwise)

III. Implementation of a mobile robot

A. AlphaBot Platform

The next step on studying the proposed strategy was the implementation of the proposed method into a real robotic platform. For that purpose, the AlphaBot platform (Fig. 2) has been used.

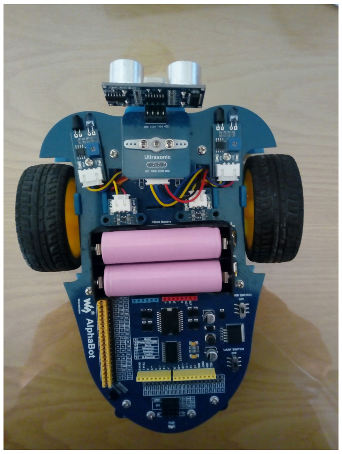

Fig. 2. The AlphaBot platform consisting of an Arduino Uno microcontroller at the bottom, an ultrasonic sensor at the top, two wheels at each side and battery components at the middle.

Fig. 3. Cell visit indication with a black line.

This platform consists of two wheels which can be rotated by two servo motors and one consisting by an omni-direction wheel. It has also an Arduino Uno microcontroller and an ultrasonic sensor that can calculate the distance of the robot from the boundaries (obstacles). The microcontroller through a USB port can be connected with the computer and programmed using the Arduino language which is similar to "C++". The ultrasonic sensor can send pulses within the range of 500 cm. During each motion command and considering that the vertical position is equal to 90 degrees, the ultrasonic sensor is rotating counter clockwise in three positions (45 degrees, 90 degrees and 135 degrees) and scans for potential obstacles. If there is an obstacle within its range then the pulses are reflected back and the sensor can detect and calculate at the same time the distance between the placed obstacles and the robot. Due to the size of the robotic platform (22cm length) it is decided that the potential dangerous distance from obstacles are within the range of 20 cm. If the distance becomes smaller than 20 cm then the robot stops its motion and the ultrasonic sensor can turn at 45 degrees counter clockwise or clockwise in order to find if there is an obstacle free path to move in. In the case where the alternatives options of movement are both forbidden then the robot moves backwards for 1 second and then considers the next motion command. In each command, it is decided that the robot can move until it has covered 30 cm distance. The size of each cell is defined as $32.5cm \times 32.5cm$ so that means, that in each command the robot can visit one cell. After that distance it stops its motion. The duration of each motion command ranges between 900ms to 1800ms due to the fact that for forward motion the robot spends the least amount of time executing this command. On the other hand, when it has to change its orientation then it spends some time turning and then moves forward. The motion strategy is presented in the flowchart in Fig. 4. Also, in order to find the areas that have been explored, a black marker has been appoint at the back of the robot that draws a straight line, as it is shown in Fig. 3.

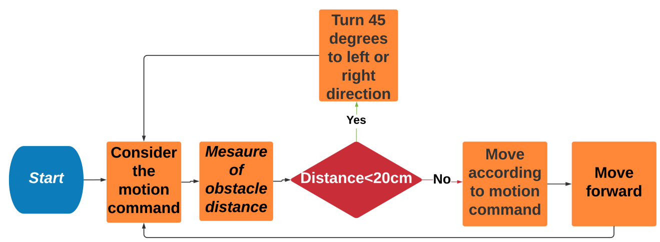

Fig. 4. Flowchart of the motion strategy

B. Experimental Results of the Proposed Method

The mobile robot starts its motion in a orthogonal workspace with boundaries and dimensions $D = 10 \times 6$ cells. There are two scenarios that have been considered. In the first one, the (10, 5) position is selected as a starting position and 120 motion commands are used. Equation (5) gives the coverage rate of the proposed method and it is computed by dividing the number of visited cells V with the overall number of cells (D).

$$Cov = \frac{V}{D} \times 100\% \tag{5}$$

Figure 5 presents the number of visits in each cell for 120 motion commands. Overall, 70% of the given space is visited by the mobile robot and it manages to avoid collision with the boundaries of the workspace. In the second scenario, the starting position changes. Now the robot starts from the position (6, 3) and it manages to cover 63.33% of the given space (Fig 6) for the same number of motion commands. For producing better results and overall understanding the behavior of the mobile robot, a mathematical analysis is held (Fig 7). In that case, the available motion commands are increasing steadily by 20 until the limit of 200 commands. As it can be noticed, the starting position plays a crucial role in the overall performance of the robot. For small number of motions, both performances are almost equal but as long as more commands are added, the difference is becoming greater.

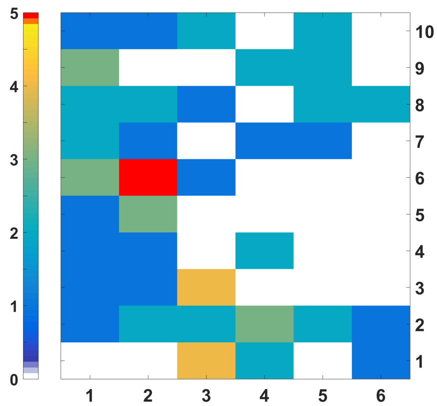

Fig. 5. Number of visits in each cell, for 120 motion commands with starting position (10, 5).

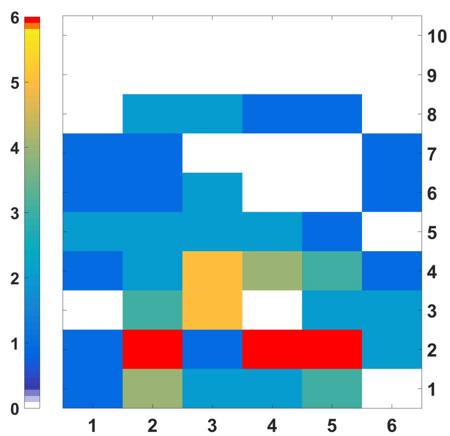

Fig. 6. Number of visits in each cell, for 120 motion commands with starting position (6, 3).

IV. Conclusion

In this work, a chaotic system was used for constructing a pseudo random bit generator. The produced bit sequence was tested by NIST 140-2 statistical package. After the evaluation of its randomness, it was used for controlling the motion of a mobile robotic system. The robot had the ability to change its motion direction in order to avoid obstacles. As a result, the robot's behavior became more natural because it could find acceptable motion paths and decided if it will follow them or not. The extensive mathematical analysis that performed, evaluated the assumption that the robot could cover larger areas as long as more motion commands were becoming available. Also, there were four basic factors that could affect the complexity of the proposed motion strategy. The change

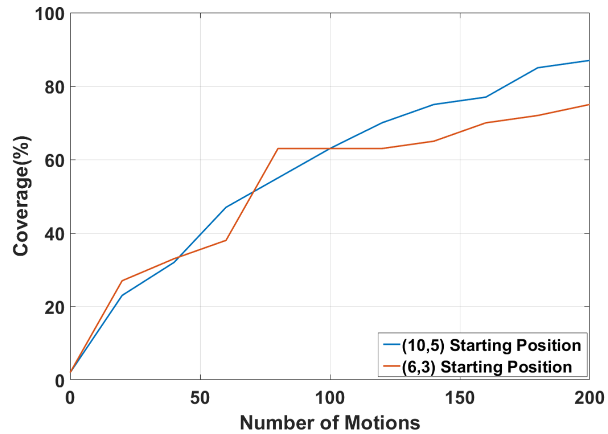

Fig. 7. Coverage performance of the proposed method for two initial positions.

in the value of either initial condition, position, parameters or orientation creates different motion trajectories and raised the complexity of the proposed method. For future work the implementation of optimized motion strategies that studied in simulations has been planned. Also, the evaluation of their success in applications in real world will be considered.

References

[1] 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 *2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, 2021, pp. 1–4.

[2] A. V. Le, P. T. Kyaw, R. E. Mohan, S. H. M. Swe, A. Rajendran, K. Boopathi, and N. H. K. Nhan, "Autonomous floor and staircase cleaning framework by reconfigurable stetro robot with perception sensors," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 1, pp. 1–19, 2021.

[3] H.-Y. Lin and Y.-C. Huang, "Collaborative complete coverage and path planning for multi-robot exploration," *Sensors*, vol. 21, no. 11, p. 3709, 2021.

[4] S. Shukla, S. Sharma, S. Kumar, V. K. Sachan, K. K. Upadhaya, and N. Shukla, "A review on the novel approach of path planning and exploration of an area for a multi mobile robotic systems," in *Recent Trends in Communication and Electronics*. CRC Press, 2021, pp. 513–518.

[5] T. Alam and L. Bobadilla, "Multi-robot coverage and persistent monitoring in sensing-constrained environments," *Robotics*, vol. 9, no. 2, p. 47, 2020.

[6] D. H. Stolfi, M. R. Brust, G. Danoy, and P. Bouvry, "Uav-ugv-umv multi-swarms for cooperative surveillance," *Frontiers in Robotics and AI*, vol. 8, 2021.

[7] H. He, Y. Cui, C. Lu, and G. Sun, "Time delay chen system analysis and its application," in *International Conference on Mechanical Design*. Springer, 2019, pp. 202–213.

[8] O. Kalayci, I. Pehlivan, A. Akgul, S. Coskun, and E. Kurt, "A new chaotic mixer design based on the delta robot and its experimental studies," *Mathematical Problems in Engineering*, vol. 2021, 2021.

[9] 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, p. 110684, 2021.

[10] GROBOTRONICS, "Alphabot," 2011-2022. [Online]. Available: https://grobotronics.com/alphabot-basic-kit.html

[11] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Booz-allen and hamilton inc mclean va, Tech. Rep., 2001.