Motion Control of a Mobile Robot Based on a Chaotic Iterative Map

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

Lazaros Moysis
Laboratory of Nonlinear Systems - Circuits & Complexity
Physics Department
Aristotle University of Thessaloniki
Thessaloniki, Greece
moysis.lazaros@hotmail.com

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

Hector Nistazakis
Department of Electronics, Computers, Telecommunications and Control
Faculty of Physics
National and Kapodistrian University of Athens, Greece
enistaz@phys.uoa.gr

Jesus Manuel Muñoz-Pacheco
Faculty of Electronic Sciences
Autonomous University of Puebla
Puebla, Mexico
jesusm.pacheco@correo.buap.mx

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

***Abstract*—In this work, the problem of designing an efficient chaotic path planning generator for exploring an area is studied. To achieve so, a Chaotic Random Bit Generator (CRBG) is first constructed. The randomness of the generator was tested and verified by the FIPS statistical package. Then, the proposed CRBG is utilized to produce the chaotic motion commands for an autonomous mobile robot moving in a 2D area, in four or eight different directions. A series of simulations have been performed in Matlab, in order to evaluate the efficiency of the proposed method and the results show that the motion in eight directions gives better results with respect to the coverage rate while also the multiple visits to the same cells have been decreased significantly.**

***Index Terms*—Chaos, Modified Logistic map, Path planning, Random bit generator, Terrain coverage**

I. Introduction

Chaos has intrigued researchers for decades since the late 60s. With the progression of technology, the study of chaos has moved on from its theoretical basis, to a plethora of different applications [1], [2]. Specifically in the area of robotics, the problem of path planning has been addressed through the use of chaotic systems [3]–[9]. The problem at hand is the design of a chaotic motion strategy for a robot moving in a 2D area, with the aim of exploring the complete area in a small number of steps, while also moving unpredictably. This combination of efficient area coverage and unpredictability can potentially be required in many applications related to exploration and surveillance. So for the achievement of both goals, chaotic systems can be applied to generate the chaotic motion, as it has been reported in the aforementioned works.

The advantage of using chaotic systems in path planning generators mainly relies on the fact that due to their high sensitivity to initial conditions and parameter values, it will be very difficult for an observer to try and replicate or predict the motion of a robot moving chaotically. For this purpose, the researchers have studied chaotic path planning methods [5]–[7], [10], [11] which can ensure the randomness of the motion trajectory and also efficiency in area coverage. The deterministic nature of the chaotic systems allow the study and comparison between the experimental and the real model.

In literature, there are two main groups of chaotic systems, continuous and discrete time. Both of them, are used in path planning and some very known are Lorenz, Chua, logistic map and others [5]–[8], [12], [13]. Many of them have been tested thoroughly and applied to path planning controllers which gave interesting results [5], [8], [10], [11], [14], [15].

In this work, a recent chaotic system is suggested for solving the problem of exploring a given workspace by a mobile robot. A Modified Logistic Map is used in combination with a simple rule, to first create a Chaotic Random Bit Generator (CRBG). Then, the effect of the de-skewing technique is studied on the randomness of the chaotic bitstream. For this purpose, the FIPS 140-2 tests suite (Federal Information Processing Standards) is used which contains a package of 15 statistical tests which verify the randomness of the produced bits. After the validation of the bit generator, the generated bit sequence is used for producing the motion commands using pairs or triads of bits. A motion in four and eight directions is considered, giving the opportunity of studying the behavior of the robot in multiple cases.

The programming environment of MATLAB was used in order to create the simulations of the proposed method. Extensive simulations were performed using various initial

values and starting positions for the robot. Also, a comparison between the two types of motion is performed. The results showed that the motion in eight directions gives better results.

The structure of the work is as follows. In Section II the appropriate chaotic random bit generator is proposed and the statistical test results are presented. In Section III the generator is applied to the chaotic path planner showing the results for both four and eight direction motions. Finally, Section IV the concludes the paper.

II. The Proposed CRBG

A. The Modified Logistic Map

Until the end of 60s researchers couldn't have the necessary tools to study properly the chaotic dynamical systems. However, when the concept of chaos was introduced in the academic community, the interest for this kind of systems soared. These systems are unique, because of their sensitivity in initial conditions, which means that even a small change in initial parameters can make the system have a completely different behavior.

In this work a modified Logistic map [16] is used, described by

$$x_{k+1} = 2\beta - x_k^2/\beta \quad , \quad k = 0, 1, 2, ... \tag{1}$$

where β is the design parameter. As it can be seen in the bifurcation diagram of the system in Fig.1(a), the system has chaotic behavior for positive values of β. The values of x_k lie in the interval $[-2\beta, 2\beta]$ and the Lyapunov exponent according to Fig.1(b) has a constant value at around 0.7, which verifies that the system is chaotic. For this work, the value of $\beta = 10^5$ is chosen.

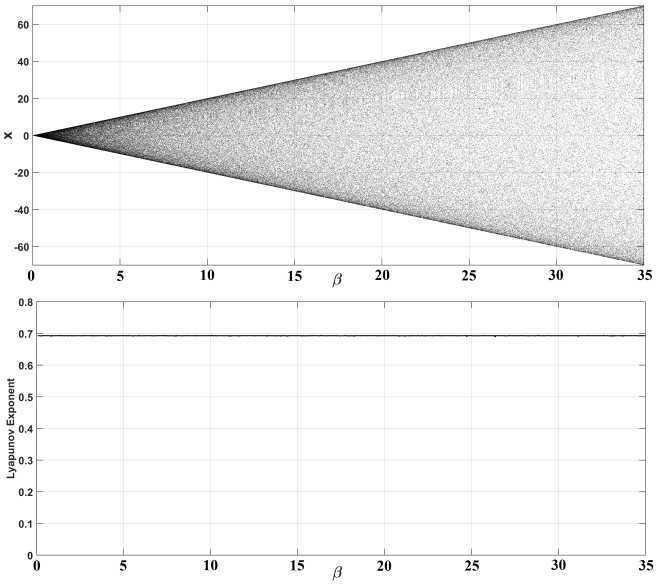

Fig. 1. (a) Bifurcation diagram of x versus the bifurcation parameter β and (b) Diagram of system's (1) Lyapunov exponent versus the parameter β.

B. Application to Bit Generation - Statistical Testing

The values of Modified Logistic map of (1) are transformed into bits according to the following procedure.

$$d_i = x_i \mod 1 \tag{2}$$

$$b_i = \begin{cases} 0 & d_i < 0.5 \\ 1 & d_i \geq 0.5. \end{cases} \tag{3}$$

So using the values of the chaotic map, the decimal part d_i is taken in each iteration and compared to the threshold value of 0.5, in order to generate a bit 0 or 1.

The next step is to reduce the correlation of the bitstream. In order to achieve that, a de-skewing technique is used which discards the pairs 00 and 11, while the 01 is converted into the output 0 and 10 into 1. Finally, for testing if the sequence is truly random, the FIPS 140-2 (Federal Information Processing Standards) tests of National Institute of Standards and Technology (NIST) are used [17]. These tests were applied for $m = 20$ groups of sequences of $1,000,000$ bits. Also, the significance level was selected as $\alpha = 0.01$. The randomness of the produced bitstream can be verified from Table I as it passes all of the 15 available tests. From the same random number sequence two kinds of bitstream have been produced. In the first one, the de-skewing technique is applied and the results are presented in the column with index P_1 and in the second with index P_2 the same technique is not used. It is clear, that the application of one more technique, such as the de-skewing, has a positive effect in the randonmness of the motion sequence.

TABLE I
FIPS 140-2 STATISTICAL TEST RESULTS

If $P \geq \alpha$ then the test is successful			
No.	**Test**	**P_1-Value**	**P_2-Value**
1	Frequency	0.534146	0.437274
2	BlockFrequency	0.739918	0.213309
3	CumulativeSums	0.090936	0.911413
4	Runs	0.213309	0.162606
5	LongestRun	0.637119	0.213309
6	Rank	0.534146	0.911413
7	FFT	0.834308	0.162606
8	NonOverlappingTemplate	0.991468	0.964295
9	OverlappingTemplate	0.534146	0.275709
10	Universal	0.911413	0.162606
11	ApproximateEntropy	0.035174	0.275709
12	RandomExcursions	0.911413	0.437274
13	RandomExcursionsVariant	0.739918	0.834308
14	Serial	0.534146	0.213309
15	LinearComplexity	0.834308	0.275709

III. Application to Chaotic Path Planning

For the application of the proposed CRBG to the problem of path planning, we consider a robot moving in a 2D space in eight or four discrete directions. Thus, the aim is to utilize the generated bit sequence to produce a set of chaotic discrete

TABLE II
BIT SEQUENCE AND MOTION COMMANDS

Motion in 4 directions	
Bits	**Motion command**
00	up
01	right
10	down
11	left

Motion in 8 directions	
Bits	**Motion command**
000	up
100	up-right
110	right
101	down-right
011	down
111	down-left
001	left
010	up-left

motion commands. These are generated using two bits at a time for motion in four directions and three bits at a time for motion in eight directions, using the rules given in Table II. So, in order to run the simulation, a bit sequence of desired length is first generated, based on the number of desired iterations to be performed. For example, for motion in eight directions and 30,000 iterations, 90,000 bits need to be generated. Then, based on the given bits, a series of motion commands are generated by reading the bitsream in sets of two or three bits at a time. In each iteration of this algorithm, if the motion, which is generated is unacceptable, that is, if it leads the robot outside the given boundaries or towards an obstacle, then the motion command is discarded and the robot awaits for the next command.

For simulation purposes, a 100×100 grid is considered, consisting of 100^2 discrete cells. As an example, Fig. 2 shows a simulation for 30,000 steps for a robot moving in four directions, starting from initial position $[50, 50]^T$. The coverage is 51.71%. The initial condition for the chaotic map is chosen randomly between $[1, 10^7]$. Also, Fig. 3 shows a colour-coded diagram depicting the number of visits in each cell.

Similarly, for a robot moving in eight directions, the simulation results for 70,000 iterations are shown in Figs. 4 and 5. The coverage here is 92.01%. Also, Fig. 6 depicts the number of visits in each cell in a color coded surface graph, which better visualizes the results.

From all of the above simulations, a question, which is raised, is how the number of motions performed affects the grid coverage, and also the average number of visits in each cell. To study this, we considered the average of five simulations starting from arbitrary initial positions, and performing a varying number of iterations each time. The simulations start from 10,000 steps and go up to 105,000, with a step size of 5,000. The results are shown in Figs. 7 and 8. Here, it is seen that motion in eight directions gives better results both in the coverage rate, and also in the number of visits in previous cells. This was expected, since diagonal motions give the robot a higher chance to visit a new cell with each iteration.

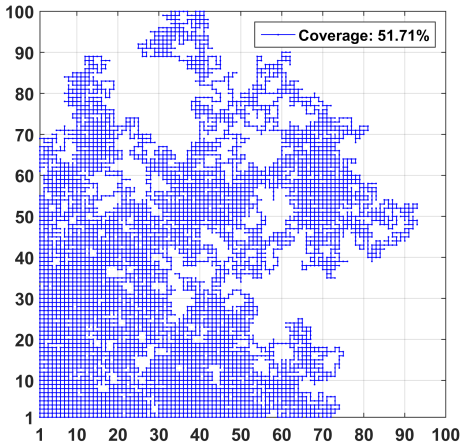

Fig. 2. Area coverage for 30,000 steps, starting from position $[50, 50]^T$, for motion in four directions.

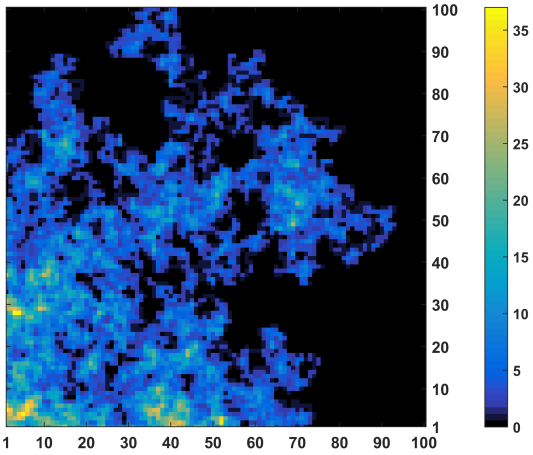

Fig. 3. Colour-coded area coverage for 30,000 steps, starting from position $[50, 50]^T$, for motion in four directions.

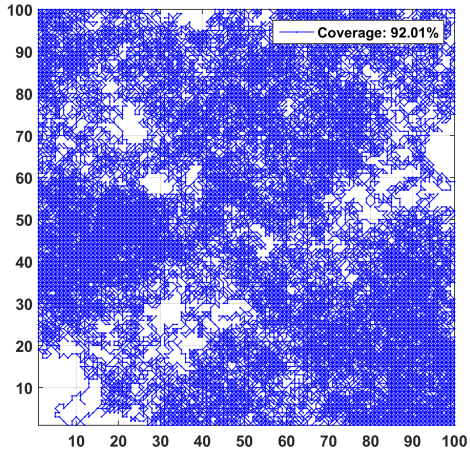

Fig. 4. Area coverage for 70000 steps, starting from position $[50, 50]^T$, for motion in eight directions.

IV. CONCLUSIONS

In this work, a chaotic random bit generator based on a modified Logistic map was constructed, in order to control the

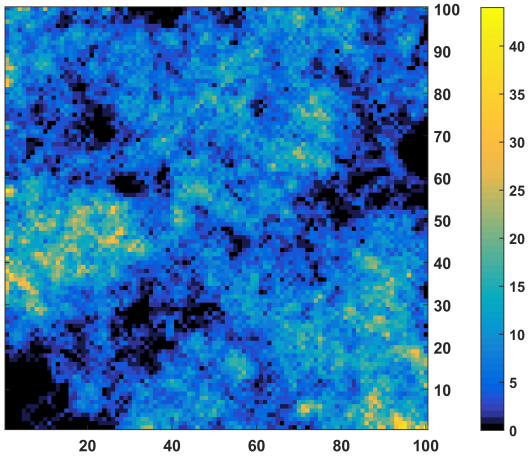

Fig. 5. Colour-coded area coverage for 70,000 steps, starting from position $[50, 50]^T$, for motion in eight directions.

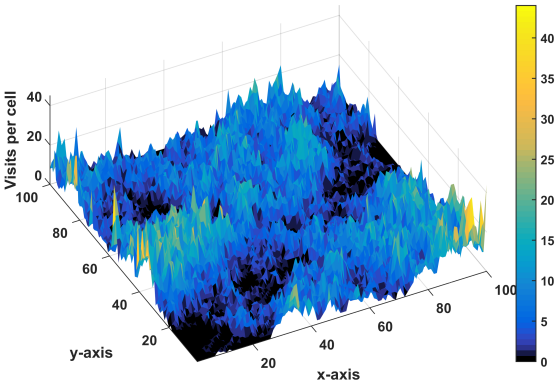

Fig. 6. 3D depiction of colour-coded area coverage for 70,000 steps, starting from position $[50, 50]^T$, for motion in eight directions.

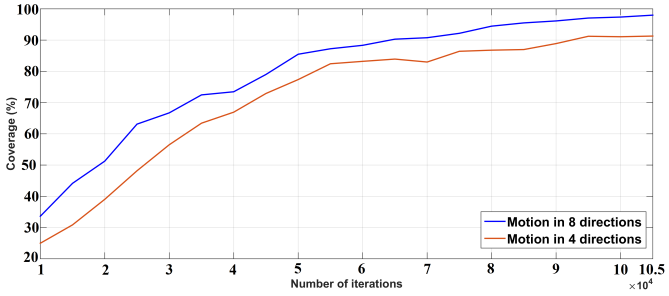

Fig. 7. Coverage stats for different number of steps.

motion of a robot. A random bitstream was created and tested by statistical package of FIPS 140-2. After the validation of the randomness, it was applied to the problem of chaotic path planning. From extensive simulations the results showed that the motion in eight direction gives superior results than in four. Future works will consider different path planning techniques which could be tested for a real robot in different environments in order to deal with potential challenges.

References

[1] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC Press, 2018.

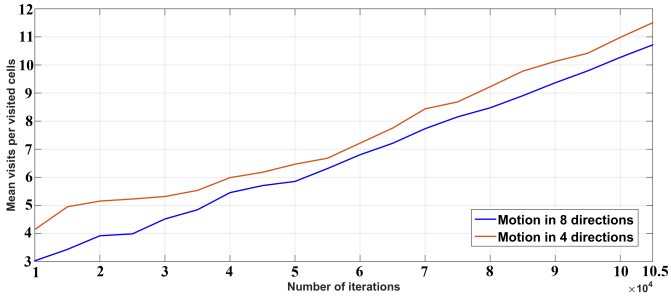

Fig. 8. Mean visits per visited cells for different number of steps.

[2] X. Zang, S. Iqbal, Y. Zhu, X. Liu, and J. Zhao, "Applications of chaotic dynamics in robotics," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 60, 2016.

[3] 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," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[4] J. Palacin, J. A. Salse, I. Valgañón, and X. Clua, "Building a mobile robot for a floor-cleaning operation in domestic environments," *IEEE Transactions on instrumentation and measurement*, vol. 53, no. 5, pp. 1418–1424, 2004.

[5] 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, 2019.

[6] 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," *International Journal of Advanced Robotic Systems*, vol. 13, no. 3, p. 107, 2016.

[7] 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 of Information Technology & Electronic Engineering*, vol. 18, no. 9, pp. 1305–1319, 2017.

[8] L. Moysis, E. Petavratzis, C. Volos, H. Nistazakis, and I. Stouboulos, "A chaotic path planning generator based on logistic map and modulo tactics," *Robotics and Autonomous Systems*, vol. 124, p. 103377, 2020.

[9] A. A. Fahmy, "Chaotic mobile robot workspace coverage enhancement," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 6, pp. 33–38, 2012.

[10] H. Choset, "Coverage for robotics - a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[11] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[12] X. Huang, L. Liu, X. Li, M. Yu, and Z. Wu, "A new pseudorandom bit generator based on mixing three-dimensional chen chaotic system with a chaotic tactics," *Complexity*, vol. 2019, 2019.

[13] V. Patidar, K. K. Sud, and N. K. Pareek, "A pseudo random bit generator based on chaotic logistic map and its statistical testing," *Informatica*, vol. 33, no. 4, 2009.

[14] L. S. Martins-Filho, E. E. Macau, R. Rocha, R. F. Machado, and L. A. Hirano, "Kinematic control of mobile robots to produce chaotic trajectories," in *Proc. of the 18th Int. Congress of Mechanical Engineering, Ouro Preto*, 2005.

[15] E. K. Petavratzis, C. K. Volos, L. Moysis, I. N. Stouboulos, H. E. Nistazakis, G. S. Tombras, and K. P. Valavanis, "An inverse pheromone approach in a chaotic mobile robot's path planning based on a modified logistic map," *Technologies*, vol. 7, no. 4, p. 84, 2019.

[16] C. Han, "An image encryption algorithm based on modified logistic chaotic map," *Optik*, vol. 181, pp. 779–785, 2019.

[17] 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.