Journal of Engineering Science and Technology Review 5 (2) (2012) 6-11

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Research Article

Motion Control of Robots Using a Chaotic Truly Random Bits Generator

Ch.K. Volos[1], I.M. Kyprianidis[2] and I.N. Stouboulos[2]

*[1]Dpt of Mathematics and Engineering Studies, Hellenic Army Academy,
Athens, GR-16673, Greece*
[2]Physics Dpt, Aristotle University of Thessaloniki, GR-54124, Greece.

Received 18 September 2011; Accepted 18 September 2012

Abstract

This work introduces chaotic motion control of mobile or humanoid robots, in order to cover a terrain. The proposed method, in order to achieve complete coverage of the entire terrain, is based on a truly random bits generator. This generator is created by a nonlinear circuit, which produces double-scroll chaotic attractors. The bits sequence, produced by the chaotic generator, is converted to a series of steps in four or eight directions. The mathematical model of the nonlinear system and the motion control strategy are described. Computer simulation tests confirm that the proposed method can obtain very satisfactory results.

Keywords: Robots, motion control, nonlinear circuits, chaos, truly random bits generator.

1. Introduction

In the last twenty years, the subject of autonomous mobile robots has acquired a keen interest because of its ever-increasing applications to various fields of economic, industrial and military activities. Furthermore, the ambition of the society to substitute humans with robots in activities, which are very sensitive to error works or in activities that put human integrity in risk, especially in military applications, has driven to the development of intelligent robotic systems. Therefore, many interesting applications of mobile robots, such as industrial transportation [1], floor-cleaning devices [2], lawn-mower vehicles and de-mining devices [3], fire fighting devices [4] and surveillance vehicles [5], have been developed.

In addition, in the last decade, the interaction between chaos theory and mobile robots has been studied intensively. It is known that the rich dynamic behavior of nonlinear systems, and especially chaotic phenomena, have been used in diverse engineering applications such as electronics [6, 7], communications [8], cryptography [9, 10], random bits generators [11] and neuronal networks [12]. This intention, to impart chaotic behavior to mobile robots, as well, is the initiation of many research activities.

The above mentioned task is achieved by designing controllers, which ensure chaotic motion. In details, signals, which are produced by chaotic systems or circuits, are used to guide autonomous robots for exploration of a terrain for vigilance, search or de-mining tasks. The high unpredictability, which is the main feature of chaotic systems, is a necessary condition in the previous mentioned tasks. Very known chaotic systems, such as Arnold dynamical system [4, 13, 14], Standard or Taylor-Chirikov map [5, 15, 16], Lorenz system [17, 18] and Chua circuit [3], have been used.

Furthermore, a very interesting task concerning autonomous mobile robots is the terrain exploration for searching (e.g. for dangerous materials, explosives) or patrolling (e.g. for intrusion in military facilities) [19-21]. For these applications many mobile robots are commercially available [22], which in many cases focus on some features such as, the perception and identification of the target, the positioning of the robot on the terrain, and the updating of the terrain's map.

However, the most important feature, determining the success of these systems, is the path planning. In this procedure the researchers try to find the way to generate a trajectory which will guarantee the finding of the explosives or the intruder. Furthermore, the path of the robot must be as much difficult to be predicted by the intruder as possible. Therefore, the problem of patrolling a terrain with a mobile robot is an issue that has to do with finding a plan for production not only of unpredictable trajectories but also a way to scan fast the entire predicted region. These are the reasons for using nonlinear dynamic systems, because the chaotic behavior of such systems ensures the unpredictability of the robot's trajectories. The fast scanning of the terrain is the subject of study among the researchers for selecting the most suitable dynamic system.

In this work, we introduce a new strategy, which generates an unpredictable trajectory, by using a chaotic truly random bits generator. The robot's motion controller produces a sequence of steps in the four basic directions (forward, right, left and backward) or in eight directions (forward, diagonal forward-right, diagonal forward-left, right, left, diagonal backward-right, diagonal backward-left, backward). In contrary with other proposed works, where the control unit defines the position goal in each step, here

* E-mail address: chvolos@gmail.com

ISSN: 1791-2377

only the motion of the mobile robot is controlled. So, the pattern and the obstacles of the proposed terrain are not of any interest.

The rest of the paper is organized as follows. The basic features of chaotic systems are presented in Section 2. The mathematical model of the nonlinear circuit, the adopted chaotic motion controller and the proposed model for the robot are described in Section 3. The simulation results and their analysis are presented in Section 4. Finally, Section 5 includes the conclusions of this work.

2. Chaotic Systems

Today, most of the robotic systems are based on microcontrollers or CPUs for controlling their chaotic motion [13]. The production of a robot's trajectory is based on the main well-known feature of the chaotic behavior, which is the great sensitivity on initial conditions.

As it is known, a dynamical system, in order to be considered as chaotic, must fulfil the following conditions [23]:

- It must be very sensitive on initial conditions,
- its chaotic orbits must be dense and
- it must be topologically mixing.

First of all, the sensitivity on initial conditions means, that a small variation on a system's initial conditions will produce a totally different chaotic trajectory. This is the feature, which is contributed to the desired robot's unpredictable path planning and makes the long-term prediction of a chaotic trajectory, based on finite-time measurements, practically impossible.

The second important feature of a chaotic system is that its chaotic orbits have to be dense. This means that, the trajectory of a dynamical system is dense, if it comes arbitrarily close to any point in the domain.

Finally, topologically mixing means that the chaotic system will move over time so that each designated area of the trajectory will eventually cover part of any particular region. This property of chaotic systems guarantees a complete scan of the whole terrain. Therefore, from the perspective of an intruder, a chaotic trajectory presents a complicated behavior, that does not exhibit any recurrent pattern and seems to be completely random. But even if the terrain coverage is similar in these two approaches, chaotic and random, chaotic has a very important advantage because it is based on determinism. This means that the behavior of a robot can be predicted in advance by the system designer. So, an autonomous mobile robot, with such characteristics, may be used successfully as a patrol robot or as a cleaning robot or as a de-mining device.

3. Chaotic Motion Generator

The chaotic truly random bits generator, which has been introduced by Yalcin, Suykens, and Vandewalle [24], is shown in Fig.1. This generator consists of four basic blocks. The first block (S_1) includes a chaotic autonomous third order circuit, which produces double scroll chaotic attractors. In the second block (S_2) the state variable x(t) is partitioned into three subspaces by the two threshold functions (Fig. 2), which are described by

$$S_2: \begin{array}{l} \sigma^1(x(\mathrm{t}))=\begin{cases}0, & \text{if } x(\mathrm{t})<\mathrm{c}_1\\ 1, & \text{if } x(\mathrm{t})\geq \mathrm{c}_1\end{cases}\\ \sigma^0(x(\mathrm{t}))=\begin{cases}0, & \text{if } x(\mathrm{t})>\mathrm{c}_2\\ 1, & \text{if } x(\mathrm{t})\leq \mathrm{c}_2\end{cases}\end{array} \tag{1}$$

The third block (S_3) is the bits generator described by the formula:

$$S_3:\ \sigma_i(\sigma^0,\sigma^1)=\begin{cases}0, & \text{if } \sigma^0=0,\sigma^1:{}_0\!\uparrow^1\\ 1, & \text{if } \sigma^1=0,\sigma^0:{}_0\!\uparrow^1\end{cases} \tag{2}$$

where, $\sigma^0:{}_0\!\uparrow^1$ shows the transition from logic 0 to logic 1 of σ^0 and $i\in\{0, 1, 2, ...\}$. Finally, the last block S_4 implements the Von Neumann's technique [25]. This technique consists of converting the bit pair 01 into the output 0, 10 into the output 1 and of discarding bit pairs 00 and 11. For testing reasons, in this paper the truly random bits generator is implemented by software. Also, it must be noticed for readers' knowledge, this is the first time, that this chaotic truly random bits generator is applied in the area of robots.

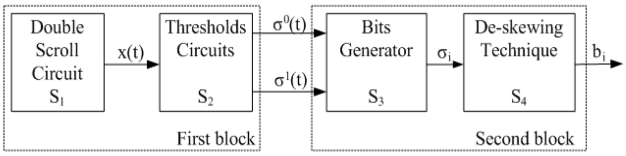

Fig. 1. The chaotic truly random bits generator.

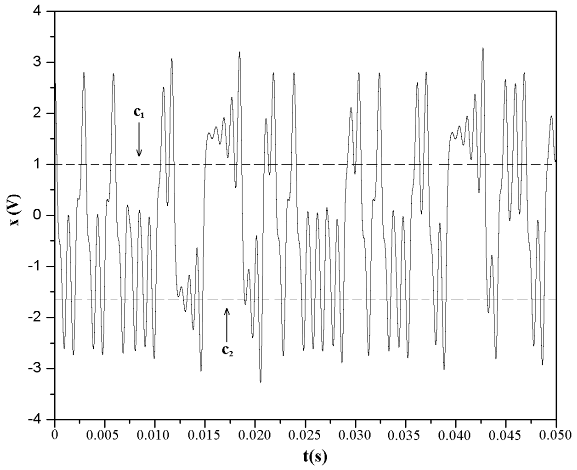

Fig. 2. Sampling the chaotic signal x(t) by using the thresholds c_1 and c_2.

3.1 The Proposed Double Scroll Chaotic Circuit

In details, the double scroll chaotic circuit, which we have designed, is shown in Fig. 3. The state equations of the circuit are the following:

$$\begin{cases} \dfrac{dx}{dt} = y \\[2ex] \dfrac{dy}{dt} = z \\[2ex] \dfrac{dz}{dt} = -\alpha \cdot x - \alpha \cdot y - \alpha \cdot z + b \cdot f(x) \end{cases} \tag{3}$$

In the system's equations, α and b are the circuit parameters and are defined as follows:

$$\alpha = (R \cdot C)^{-1}, \quad b = (R_X \cdot C)^{-1} \tag{4}$$

Also, *x*, *y*, and *z* are the state variables, which represent the voltages at the outputs of the operational amplifiers numbered as "1", "2" and "3" respectively, as shown in Fig. 3.

Function f(*x*) in systems equation (3) is a saturation function, which represents the voltage at the output of the operational amplifier numbered as "5" and is defined by the following expression:

$$f(x) = \begin{cases} 1, & \text{if} \quad x > \dfrac{R_2}{R_3} \cdot 1\,V \\[2ex] \dfrac{R_3}{R_2} \cdot x, & \text{if} \quad -\dfrac{R_2}{R_3} \cdot 1\,V \le x \le \dfrac{R_2}{R_3} \cdot 1\,V \\[2ex] -1, & \text{if} \quad x < -\dfrac{R_2}{R_3} \cdot 1\,V \end{cases} \tag{5}$$

So, f(*x*) is implemented in such a way that the saturation plateaus are ±1 and the slope of the intermediate linear region is $n = R_3/R_2$.

The values of the circuit elements were: R = 20 kΩ, R_1 = 1 kΩ, R_2 = 14.3 kΩ, R_3 = 28.6 kΩ, R_X = 12.5 kΩ and C = 1 nF. Consequently, α = 0.5 and b = 0.8. Furthermore, the voltages of the positive and negative power supplies were set ±15 V and the operational amplifiers were of the type LF411. In this work, the system (3) was solved numerically by using the fourth order Runge-Kutta algorithm. For the above set of parameters and for the chosen set of initial conditions ($x_0 = 0.2$, $y_0 = 0.4$, $z_0 = 0.5$), the Lyapunov exponents were calculated: $LE_1 = 0.17118$, $LE_2 = 0$, $LE_3 = -0.97564$. According to the theory of nonlinear systems, the existence of one positive Lyapunov exponent confirms numerically the appearance of the chaotic attractor (Fig. 4) of the proposed circuit.

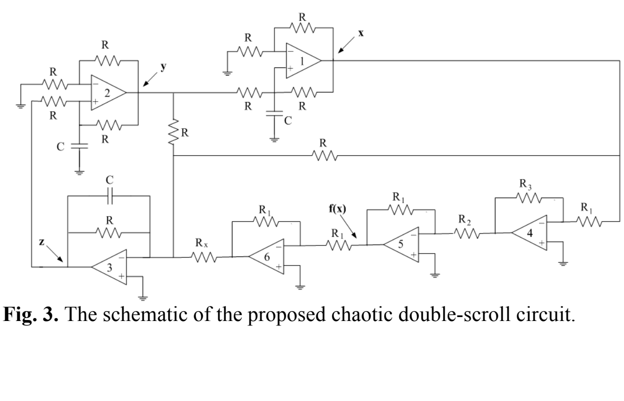

Fig. 3. The schematic of the proposed chaotic double-scroll circuit.

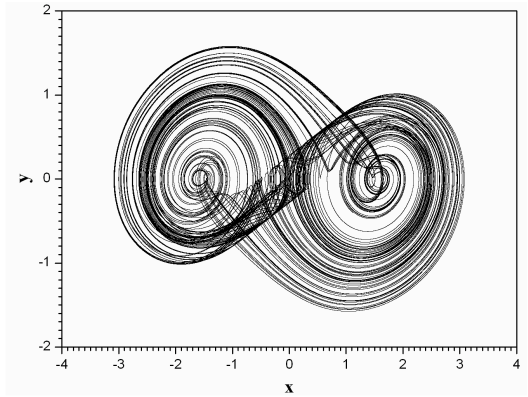

Fig. 4. The chaotic double-scroll attractor of the proposed nonlinear circuit.

3.2 Statistical Tests of the Chaotic Bits Generator

The "randomness" of the produced bits sequence by the proposed chaotic generator is confirmed by using one of the most important statistical test suites, the FIPS (Federal Information Processing Standards, FIPS-140-2) of the National Institute of Standards and Technology (NIST) [26]. In this paragraph the results of the use of the FIPS's statistical tests, such as Monobit test, Poker test, Runs test, and Long run test, are presented.

The least number of sequence's bits, which are produced by the chaotic truly random bits generator, must be 20000 bits. So, based on FIPS-140-2, the bitstream of the proposed chaotic generator must satisfy the following standards.

- Monobit Test: The number n_1 of 1's in the bitstream must be $9725 < n_1 < 10275$.
- Poker Test: This test determines whether the sequences of length n (n = 4) appear approximately the same number of times in the bitstream. The bounds of this statistic are then $2.16 < X_3 < 46.17$.
- Runs Test: This test determines whether the number of 0's (Gap) and 1's (Block) of various lengths in the bitstream are as expected for a random sequence.
- Long Run Test: This test is passed if there are no runs longer than 26 bits.

The values c_1 and c_2 of the two thresholds of the second block (S_2) of the generator are set to 1.0 V and –1.68 V, respectively. For these values of c_1 and c_2 we have calculated the measure-theoretic entropy [27] of the proposed truly random bits generator, which are $H_n = 0.6914$ for n = 3 and $H_n = 0.6921$ for n = 4, where n is the length of the n-word sequences. In Table1 the results of specified tests of FIPS-140-2, which confirm that the proposed TRBG has passed all the statistical tests, are shown.

Table 1. Results of FIPS-140-2 test, for the chaotic truly random bits generator

Monobit Test	Poker Test	Runs Test	Long Run Test
n_1=10015 (50.075%)	8.4289	B_1=2605 B_2=1183 B_3=651 B_4=347 B_5=139 B_6=158	No
Passed	Passed	Passed	Passed

3.3 The mobile Robot

A great number of works on kinematic control of chaotic robots is based on a typical differential motion with two degrees of freedom, composed by two active, parallel, and independent wheels and a third passive wheel. However, the proposed, in this work, robot model represents an interesting compromise of simplicity between control and implementation in many robot models. As it is mentioned, the chaotic motion controller produces a sequence of steps in the four basic directions (forward, right, left and backward) or in eight directions (forward, diagonal forward-right, diagonal forward-left, right, left, diagonal backward-right, diagonal backward-left, backward). This is implemented by converting the bits pairs: 00, 01, 10 and 11, produced by the chaotic generator, into steps in the following direction: forward, right, left and backward. With the same way the bits triads: 000, 001, 010, 011, 100, 101, 110 and 111, are converted into steps in the following directions: forward, diagonal forward-right, diagonal forward-left, right, left, diagonal backward-right, diagonal backward-left and backward. So, this kinematic control model is very easy to be implemented not only in differential motion robots (Fig. 5(a)), in which the robot rotates around itself followed by a linear displacement directly to the next position, but especially in humanoid robots, (Fig. 5(b)), which will have a rapid growth of interest in the upcoming decade.

Also, in the real world, robots move in spaces with boundaries like walls or obstacles. Furthermore, many robots have sensors, like sonar or infrared devices, which provide the capability to detect the presence of obstacles or even more the recognition of the searched objects or intruders. In this work, for a better understanding of the behavior of the robot's chaotic motion generator, we assume that the robot works in a smooth state space with boundaries and without any sensor. So, in the case that the proposed robot reaches boundaries (like borders) of the terrain, two different approaches are adopted. In the first case the robot reaches the boundary and waits the next direction order to move, while in the second case a mirror mapping-like approach is followed.

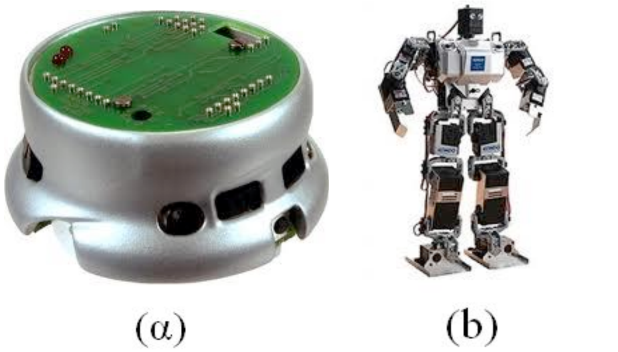

Fig. 5. (a) The mobile robot Khepera and (b) the humanoid robot Kondo KHR-2HV.

4. Numerical Simulations of the System

In the following step of validation tests, we have simulated numerically the robot kinematic motion, by analyzing the terrain coverage, using the known coverage rate (C), which represents the effectiveness, as the amount of the total surface covered by the robot running the algorithm. For this reason we define a square terrain with dimensions $M = 25 \times 25 = 625$ in normalized unit cells. Then the coverage rate (C) is given by the following equation

$$C = \frac{1}{M} \cdot \sum_{i=1}^{M} I(i) \tag{6}$$

where, I(i) is the coverage situation for each cell [28]. This is defined by the following equation

$$I(i) = \begin{cases} 1, & \text{when the cell i is covered} \\ 0, & \text{when the cell i is not covered} \end{cases} \tag{7}$$

where, i = 1, 2, ..., M. Also, another interesting evaluation criterion is the coverage time of the system, which is the total number of steps (time) for the system to cover the whole terrain.

The simulation starts from an arbitrary initial position, which does not play any crucial role in coverage rate, as it will be analyzed thoroughly in the next paragraph. Based on the minimum number of produced bits (n = 20000) for testing its reliability with the above mentioned statistical tests (FIPS-140-2), the proposed motion controller produced 10000 steps in the first case (motion in four directions) and 5000 steps in the second case (motion in eight directions). In order to accomplish our goal, the complete terrain coverage, the chaotic robot must move in every region, covering systematically the entire terrain.

4.1 First Case

As it is mentioned, in the first case, if the robot reaches the terrain's boundaries waits the next direction order to move. This may cause a little delay in terrain's coverage and a loss of some steps. But these disadvantages are set aside because of the convenience of the robot's kinematic control law.

So, in the following example the motion generator produces a sequence of steps in the four basic directions (forward, right, left and backward) starting from three different initial positions on the terrain: $(x_0, y_0) = \{(13, 1), (13, 13), (20, 3)\}$. The results for 2000 and 10000 robot's steps for $(x_0, y_0) = (13, 1)$, are shown in Fig. 6. Especially, in Fig. 6(b) the coverage of the whole terrain can be observed. All the cells of the terrain have been visited by the robot, except of only five cells in two terrain's corners. In Fig. 7, the coverage rate versus the number of steps is shown, for the robot with the proposed chaotic motion generator, starting from the three, above mentioned, initial positions. In the three simulations, the same final terrain's coverage percentage (99.2%) was calculated. Also, the plots of the coverage rate versus the number of steps in Fig. 7 have the same form, regardless of the three different initial positions. Furthermore, the robot has covered, practically, all the terrain (94.4%) after the 4200th step. The next steps are necessary for covering the rest terrain, approximately the 5% of the coverage area. So, the graphs in Fig. 7 allow us to conclude, that the complete terrain's coverage of the proposed robot's chaotic motion generator is ensured.

Next, a different approach of the motion control, when the robot reaches the borders of the terrain, is presented. When the robot meets one of the terrain's borders, instead of waiting the next step order from the generator, it moves in an opposite direction. For example, if the robot has reached the right border and has an order to move right, which is unacceptable, it finally moves one step to the left.

As we can see in Fig. 8, the robot has covered almost the whole terrain with this second motion control approach, for an arbitrary initial position $(x_0, y_0) = (20, 5)$. There are only 11 unit cells, which have not been scanned from the robot. However, the comparison with the first approach proves, that the second approach has 6 more uncovered unit cells than the first one (11 instead of 5). This becomes more obvious from the plot of the coverage rate versus the number of steps for the two proposed approaches (Fig. 9). The coverage rate is increased with the same way in the two approaches until almost the 4000th step (86.88% and 86.40% respectively), while in the next 6000 steps the robot tries to coverage the whole terrain (99.20% and 96.96%). So, the second approach has a little disadvantage in regard to the first approach relatively to the coverage rate of the terrain.

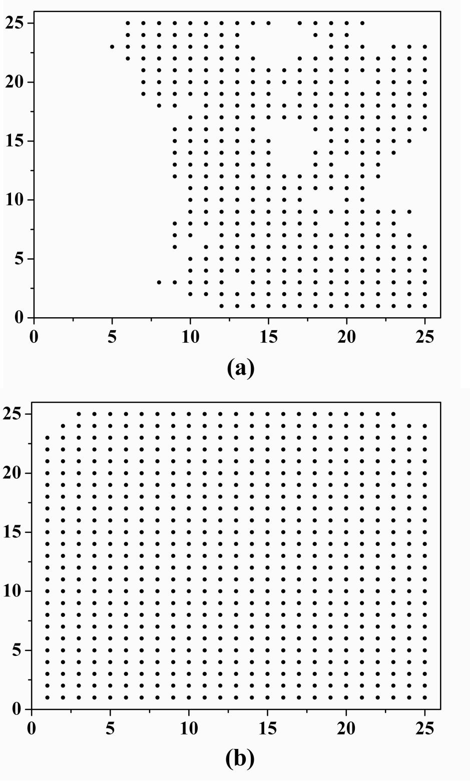

Fig. 6. Terrain covering using the robot with the proposed chaotic generator in the first case for initial position $(x_0, y_0) = (13, 0)$, for (a) 2000 steps and (b) 10000 steps.

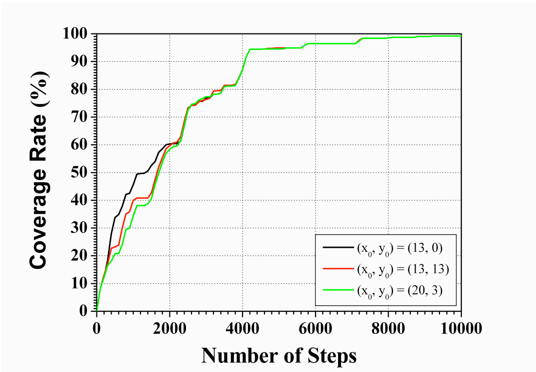

Fig. 7. Coverage rate versus number of steps, for the robot with the proposed chaotic generator, for three different initial positions, in the first case.

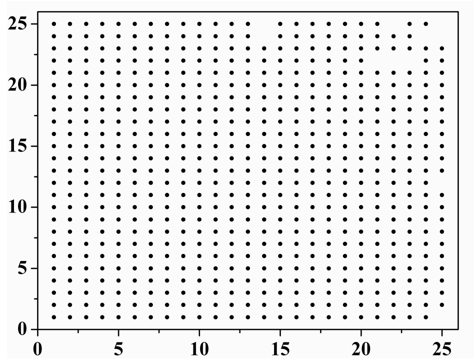

Fig. 8. Terrain covering using the robot with the proposed mirror mapping like technique for initial position $(x_0, y_0) = (20, 5)$.

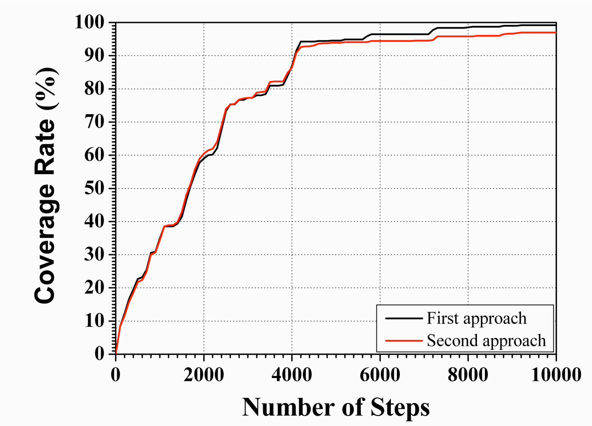

Fig. 9. Coverage rate versus number of steps, for the robot with the proposed chaotic generator, for the two motion control approaches in the first case.

4.2 Second Case

In this paragraph the numerical simulation results of the robot's kinematic motion by using the capability of the robot to move in eight directions (forward, diagonal forward-right, diagonal forward-left, right, left, diagonal backward-right, diagonal backward-left, backward), are presented. The coverage rate was used, once again, in order to compare the effectiveness of this case, as the total surface is covered by the robot running the algorithm, in regard to the previous one. Furthermore, the same technique for the robot's control has been applied. So, if the robot reaches the terrain's border waits the next direction order.

For the validation of this approach an arbitrarily initial position is chosen $(x_0, y_0) = (2, 18)$. In Fig. 10, the coverage rate versus the number of steps, for 5000 steps in this case, in comparison to the previous one is shown. In general, the robot shows a more quick coverage of the terrain's space until 2000th step in the case of moving in four directions. For this number of steps the robot has covered, in both cases, approximately the 60% of the terrain. In the remaining 3000 steps the plots are almost the same and finally the total terrain's coverage percentage was calculated to be equal to 95.84% in both cases. Therefore, the robot, with the capability of moving in eight directions, has the same total coverage rate with that of the previous mentioned case. But it has no so good performance at the beginning of the terrain's scanning.

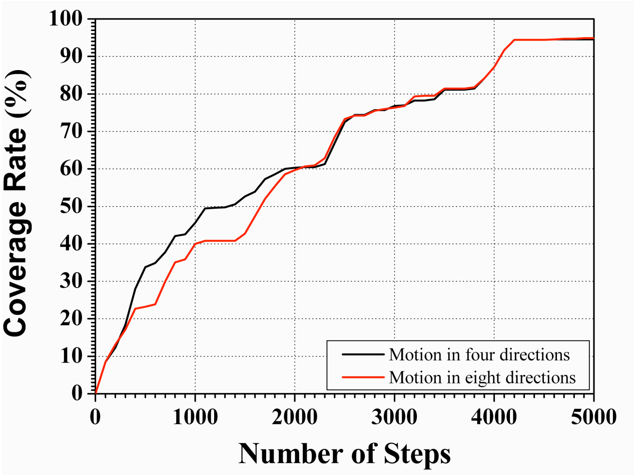

Fig. 10. Coverage rate versus number of steps, for the robot with the proposed chaotic generator, for the two kinematic control cases.

5. Conclusion

In this work, the implementation of a chaotic robot, which implies a robot with a controller that guarantees a chaotic motion, is presented. Unlike previous works, in which the controller defines the position goal in each step by imparting a chaotic motion behavior, here the robot's motion controller produces a sequence of steps in the four basic directions or in eight directions. For this reason a chaotic truly random bits generator, consisting of a chaotic double-scroll circuit, was used.

Validation tests, based on numerical simulations of the robot's motion control, confirm that the proposed method can obtain very satisfactory results. Especially, the case of moving in four directions, regardless of the simplicity in implementation, provides better results. Also, the proposed approach has several interesting features and advantages. Firstly, this strategy ensures high unpredictability of robot trajectories, resembling a non-planned motion from external observer's point of view. It is also important to mention, that the proposed strategy has an advantage over a random walk-like motion, which is the deterministic nature of this chaotic control law.

Finally, the results of this work ensure that the application of nonlinear systems, and especially chaotic circuits, to solutions for robots control strategies represents a very interesting task for researchers of both scientific fields. Furthermore, this is our first attempt to engage with this subject, while the results of the proposed robot's motion command show that such strategy can be easily applied in real robots. So, in future works, an experimental realization of the discussed idea will be studied.

References

1. J.H.Suh, Y.J.Lee, and K.S.Lee, J. Mech. Sc. Techn. 19, 1720 (2005).
2. J.Palacin, J.A.Salse, I.Valganon, and X.Clua, IEEE Trans. Instrum. Meas. 53, 1418 (2004).
3. P.Sooraksa, and K.Klomkarn, IEEE Circuits Syst. Mag. 10, 46 (2010).
4. M.J.M.Tavera, M.S.Dutra, E.Y.V.Diaz, and O.Lengerke, Proc. 20th Int. Congress Mechanical Engineering, Gramado, Brazil, (2009).
5. L.S.Martins-Filho and E.E.N.Macau, Studies in Computational Intelligence, Springer-Verlag Berlin Heidelberg, vol. 76, 109 (2007).
6. M.P.Hanias, J. Eng. Sci. Techn. Rev. 1, 33 (2008).
7. I.M.Kyprianidis, Ch.K.Volos, S.G.Stavrinides, I.N.Stouboulos, and A.N.Anagnostopoulos, J. Eng. Sci. Techn. Rev. 3, 41 (2010).
8. S.G.Stavrinides, A.N.Anagnostopoulos, A.N.Miliou, A.Valaristos, L.Magafas, K.Kosmatopoulos, and S.Papaioannou, J. Eng. Sci. Techn. Rev. 2, 82 (2009).
9. Ch.K.Volos, I. M.Kyprianidis, and I.N.Stouboulos, WSEAS Trans. Circ. Syst. 5, 1654 (2006).
10. Ch.K.Volos, I.M.Kyprianidis, and I.N.Stouboulos, Int. J. Multimedia Intelligence and Security 1, 320 (2010).
11. M.E.Yalcin, A.K.Suykens, and J.Vandewalle, IEEE Trans. Circuits Syst. I 51, 1395 (2004).
12. M.Ebner, and S.Hameroff, Proc. 3rd Int. Conf. Biocomputational Systems and Biotechnologies, Venice/Mestre, Italy, pp. 67-72 (2011).
13. Y.Nakamura, and A.Sekiguchi, IEEE Trans. Robot. Autom. 17, 898 (2001).
14. A.Jansri, K.Klomkarn, and P.Sooraksa, Proc. 30th IEEE Annual Conf. Industrial Electronics Society, Busan, Korea, vol. 3, pp. 2536-2541 (2004).
15. L.S.Martins-Filho, E.E.N.Macau, R.Rocha, R.F.Machado, and L.A.Hirano, Proc. 18th Int. Congress of Mechanical Engineering, Ouro Preto, Brazil, (2005).
16. L.S.Martins-Filho and E.E.N.Macau, Math. Probl. Eng. 2007, 1 (2007).
17. L.S.Martins-Filho, R.F.Machado, R.Rocha, and V.S.Vale, Symposium Series in Mechatronics 1, 40 (2004).
18. D.I.Curiac, and C.Volosencu, Proc. 10th WSEAS Int. Conf. Automation and Information, Prague, Czech Republic, pp. 366-369 (2009).
19. E.Krotkov, and J.Blitch, Int. J. Rob. Res. 18, 769 (1999).
20. H.R.Everett, IEEE Instrum.Meas. Mag. 6, 30 (2003).
21. F.Capezio, A.Sgorbissa, and R.Zaccaria, Proc. 5th IEEE Int. Workshop on Robot Motion and Control, Dymaczewo, Poland, pp. 157-162 (2005).
22. www.MobileRobots.com.
23. B.Hasselblatt, and A. Katok, A First Course in Dynamics: With a Panorama of Recent Developments, Cambridge University Press, Cambridge (2003).
24. M.E.Yalcin, A.K.Suykens, and J.Vandewalle, IEEE Trans. Circuits Syst. I 51, 1395 (2004).
25. J.Von Neumann, Various Techniques Used in Connection with Random Digits, Forsythe, G.E. (Ed.), Applied Mathematica Series-Notes, vol. 12, pp.36-38, National Bureau of Standards (1951).
26. http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.
27. A.M.Fraser, IEEE Trans. Inf. Theory 35, 245 (1989).
28. S.Choset, Ann. Math. Artif. Intel. 31, 113 (2006).