Journal of Engineering and Sustainable Development
http://jeasd.uomustansiriyah.edu.iq
https://doi.org/10.31272/jeasd.24.4.6

Vol. 24, No. 04, July 2020
ISSN 2520-0917

MOBILE ROBOT MOTION CONTROL BASED ON CHAOTIC TRAJECTORY GENERATION

*Ammar Ibrahim Majeed

Lecturer, Department of Electrical Engineering, College of Engineering, Mustansiriyah University، Baghdad, Iraq

Received 9/4/2019 ***Accepted in revised form 30/10/2019*** ***Published 1/7/2020***

Abstract: This paper introduces chaotic motion control of autonomous differential drive wheeled mobile robot to cover a certain surrounded area completely and as fast as possible. The suggested method was based on utilization of chaotic logistic map to accomplish a complete coverage of entire area. The trajectory is created by a microcontroller and fed to the designed robot using the proposed chaotic trajectory generator. The designed robot was occupied with infrared sensors to avoid the unforeseen obstacles during the navigation. The mathematical model of the designed robot and the motion control strategy were described in details. Computer simulation tests as well as the practical tests by using the designed robot show satisfactory results and confirm the success of the proposed method.

Keywords: *Wheeled mobile robot, Terrain coverage, Motion control, Microcontroller, Chaos.*

1. Introduction

Wheeled mobile robot (WMR) has witnessed a considerable interest in the last decades. It was used in many applications within unstructured environments where an autonomous mobile robot is necessary.

Path planning represents a fundamental importance for the WMR. Terrain covering task, which is a special case of path planning, requires the robot to cover all partitions of a specific workspace in a fast and complete manner. Along with area cleaning robots, lots of robotic applications require terrain covering task, e.g., demining, lawn mowers, painting, autonomous underwater covering, agricultural crop harvesting equipment, and windows cleaning robots [1]. The terrain covering task should be accurate and fast as possible with ability of obstacle avoidance.

Many approaches were studied to accomplish terrain covering task such as, approximate, semi approximate and exact cellular decomposition, fuzzy logic, artificial potential field, template based model, and neural networks. Most of these approaches assume static environment whereas covering real world requires a safe movement, without collisions, in unstructured environments and cover the entire workspace with minimum time elapsed.

In this paper a chaotic logistic maps were utilized to control the robot motion during terrain covering task of an autonomous WMR.

2. Kinematics System Control

The mobile robot that designed through this research is an ordinary differential drive mobile robot with two degree-of-freedom and it was occupied with two front independently driven wheels, a single rear, passive free steered wheel (caster wheel) which is used for equilibrium purposes, and five infra-red (IR) proximity sensors used for obstacles avoidance purposes.

**Corresponding Author*: ammar.alkindy@uomustansiryah.edu.iq

The control of the two front wheels comprises velocity and rotation control and it was performed by the microcontroller. The infra-red (IR) sensors were used to perform obstacles avoidance by measuring the distances to these obstacles. The designed robot can be occupied with other type of sensors to detect and recognize the explored objects.

The robot body was assumed as rigid body moving within a horizontal x-y plane. The robot motion was accomplished through driving the two active wheels and there is no slip occurs between the wheels and the floor (no side skid). The movement of the robot is represented by instantaneous linear velocity v(t) [m/s] and instantaneous orientation θ(t) [rad] of point P that lies in the middle of the active wheels axis. The rotation velocity of the robot body at point P was represented by ω(t) [rad/s]. The posture of the mobile robot in Cartesian coordinate system is shown in Fig. 1.

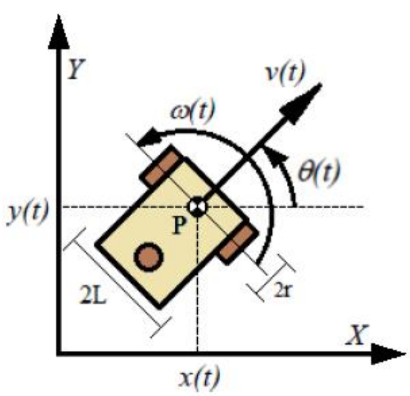

Figure 1. Posture of the mobile robot in x-y plane.

To control the robot motion, the wheels linear velocities, $v_{Left}(t)$ and $v_{Right}(t)$ or similarly robot linear and angular velocities v(t) and ω(t) respectively, (named the input variables) should be provided. Mathematical modeling of the designed robot can be represented by the following equations [2]:

$$\dot{x}_c(t) = v(t)cos\theta(t) \tag{1}$$

$$\dot{y}_c(t) = v(t)sin\theta(t) \tag{2}$$

$$\dot{\theta}(t) = \omega(t) \tag{3}$$

The robot posture in the inertial Cartesian frame {x,y,θ} can be specified by using the vector $\boldsymbol{q}$=$[x\ y\ \theta]^T$.

The kinematic equations (1), (2), and (3) can be represented in matrix form as follows [3]:

$$\dot{\boldsymbol{q}} = \boldsymbol{S}(\boldsymbol{q})\boldsymbol{V}(\boldsymbol{t}) \tag{4}$$

$$\dot{\boldsymbol{q}} = [\dot{\boldsymbol{x}}\ \dot{\boldsymbol{y}}\ \dot{\boldsymbol{\theta}}]^{\boldsymbol{T}} \tag{5}$$

$$\boldsymbol{V} = [\boldsymbol{v}\ \boldsymbol{\omega}]^{\boldsymbol{T}} \tag{6}$$

$$\boldsymbol{S}(\boldsymbol{q}) = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \tag{7}$$

The angular velocities of the left and right wheels can be used to calculate the linear velocities of these wheels as follows:

$$V_{Left}(t) = r\ \omega_{Left} \tag{8}$$

$$V_{Right}(t) = r\ \omega_{Right} \tag{9}$$

In these equations, r represents the radius of the active wheels. ω_{Left} and ω_{Right} are angular velocities of the left and right wheels. The total linear and angular speed of the robot can be calculated as follows:

$$v_R = \frac{(v_{Right}+v_{Left})}{2} \tag{10}$$

$$\omega_R = \frac{d\theta}{dt} = \frac{(v_{Right}-v_{Left})}{d} \tag{11}$$

The posture and direction of the robot can be estimated as follows:

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{(v_{Right}+v_{Left})}{2} \\ \frac{(v_{Right}-v_{Left})}{d} \end{bmatrix} \tag{12}$$

3. Dynamic Control

To deduce the dynamic behavior of the designed robot, the following dynamic equation was used [3]:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{V}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{\tau}_d = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{\tau} - \boldsymbol{A}^T(\boldsymbol{q})\boldsymbol{\lambda} \quad (13)$$

Where, $\boldsymbol{M}(\boldsymbol{q})$ denotes the inertia matrix, $\boldsymbol{Vm}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is the centripetal and Coriolis matrix, $\boldsymbol{F}(\dot{\boldsymbol{q}})$ represents the surface friction, $\boldsymbol{G}(\boldsymbol{q})$ denotes the gravitational vector, $\boldsymbol{\tau_d}$ represents the disturbances and unstructured dynamics, $\boldsymbol{B}(\boldsymbol{q})$ denotes the input transformation matrix of the input, $\boldsymbol{\tau}$ represents the input vector, $\boldsymbol{A}(\boldsymbol{q})$ is the constraints matrix, and $\boldsymbol{\lambda}$ represents the constraint forces vector.

Mathematically and from equations (4) and (13), the complete motion equations for differential drive robot can be obtained and as follows:

$$\dot{\boldsymbol{q}} = \boldsymbol{S}(\boldsymbol{q})\boldsymbol{V}(\boldsymbol{t}) \quad (4)$$

$$\boldsymbol{S^T M S \dot{V}} + \boldsymbol{S^T}\left(\boldsymbol{M\dot{S}} + \boldsymbol{V_m S}\right)\boldsymbol{V} + \boldsymbol{s^T \tau_d} = \boldsymbol{S^T B \tau} \quad (14)$$

Where:

$\dot{\boldsymbol{V}}$ is the mobile robot's acceleration vector, which can be represented by the linear acceleration $\dot{v}$ and angular acceleration $\dot{\omega}$ as follows:

$$\dot{\boldsymbol{V}} = [\boldsymbol{v}^{\cdot} \ \ \boldsymbol{\omega}^{\cdot}\ ]^{\boldsymbol{T}} \quad (15)$$

4. Chaotic Trajectories

Chaos represents one of the most important phenomena in the nonlinear systems. During this class of behaviors, the systems would exhibit special features including unpredictability, continuous frequency spectra and repetition tendency.[٤]

The logistic map represents one of the discrete chaos systems. It can be used to model a chaos behavior generation. The simplicity of this map as compared with other chaotic systems pushes the researchers to depend on it in robotics. The logistic map equation can be represented mathematically as follows [5,6]:

$$X_{n+1} = \mu X_n(1 - X_n) \quad (16)$$

In this equation μ lies in the range [0,4], n= 0,1,2... represents the discrete time intervals. This chaos series would start from the initial value x_0, and the rest values of the chaotic series would be obtained from iteration to the equation (16). Although the initial value x_0 is different slightly, the generated chaotic series show comparable chaos features when the factor μ changes within the interval (3.57-4]. Actually, x_0 values within the interval (0-1) become increasingly chaotic when μ converges to 4 [7]. The discrete form of the equations (1-3) is:

$$\left.\begin{aligned} x_{n+1} &\approx x_n + hvcos\theta_n \\ y_{n+1} &\approx y_n + hvsin\theta_n \\ \theta_{n+1} &\approx \theta_n + h\omega_n \end{aligned}\right\} \quad (17)$$

Where h was chosen to be very small step.

The range of variation of the robot orientation angle θ_n lies within the range (-π/2,+π/2) and the values of x_n for the logistic map are lies in the range (0-1). To integrate the chaotic map with the path generator of the mobile robot, the following transformation formula should be used:

$$\left.\begin{aligned} \theta_n &= \pi x_n - \pi/2 \\ x_{n+1} &= 4x_n(1 - x_n) \end{aligned}\right\} \quad (18)$$

Therefore the robot discrete model would be as follows:

$$\left.\begin{aligned} x_{n+1} &\approx x_n + hvcos\theta_n \\ y_{n+1} &\approx y_n + hvsin\theta_n \\ \theta_n &= \pi x_n - \pi/2 \\ x_{n+1} &\approx 4x_n(1 - x_n) \end{aligned}\right\} \quad (19)$$

From these equations it can be shown that the angle θ_n of the robot is computed iteratively using equations (19). Therefore the robot path generator would be dependent on equations (19) that have the chaotic features.
The area coverage could be judged using the performance index k which is defined as the ratio of the area passed by the robot A_p to the total workspace area A_t.

$$k = A_p/A_t \tag{20}$$

The passed area A_p and the total area A_t can be calculated by the following algorithm:
- Dividing the whole area into (N*M) pixels (i.e., A_t).
- Using image processing to assign to 255 for all white pixels and 0 for all black pixels which pass through the trajectory of the robot.
- Count the number of 255 pixels W (i.e., white pixel).
- Then passed area A_p= A_t - W = N*M – W.

The properties of terrain coverage can be analyzed by simulation considering the basic mission requirements for fast terrain scanning.

5. Simulation Program

To investigate the system model represented by equations (19), a simulation program for mobile robot is needed. This simulation program must provide a virtual workspace with modification capability, such as changing the robot size and position and the obstacles numbers, size and position [8]. Many simulation programs can be used such as KiKS, Wibot, … etc. The KiKs simulator has been chosen to perform the simulation part of this work because it has a virtual mobile robot called Khepera qualified the requirements of this work and can operate under Matlab program. The simulation environment of KiKs program is shown in Fig. 2 below.

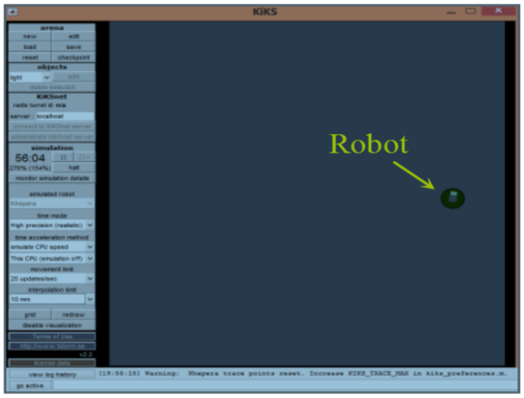

Figure 2. Simulation environment of KiKs program.

6. Robot Construction

The structure of the designed mobile robot is shown in Fig. 3. It consists of two front differential drive active wheels, one in each side of the robot body. These wheels are driven by a separate DC motor individually. These motors are controlled by using PWM signals generated by the PIC18F45K22 Microcontroller with the help of a driver integrated circuit SN 754410. Furthermore, a caster wheel was used to ensure robot body balance during mobilization. Obstacles avoidance has been done through measuring the distances to the wall and the obstacles using five IR proximity sensors from Sharp.

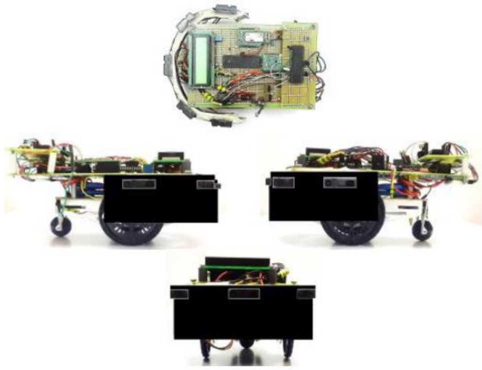

Figure 3. The physical construction of the designed robot.

7. Simulation Results

The simulation process was carried out using the proposed chaotic path planner and applying the mobile robot movement equations represented by equations (19). Many environments for the workspace of the mobile robot were investigated including square, square divided into two sections with narrow tunnel, square with two obstacles in the middle, and square with four obstacles at corners. Furthermore, a comparison between the proposed system and the Arnold system is accomplished. Examples of trajectories of the robot are shown in Figs. (4-7). Where the chaotic mobile robot run at following condition:

Linear velocity: v = 0.15 m/s

Initial states: [x y θ]T = [4 4 π/2]T

Work space limits: 1000 x 1000 mm

Simulation interval: 5000 sec

Evaluating the chaotic controller by means of performance index k is shown in Fig. 8.

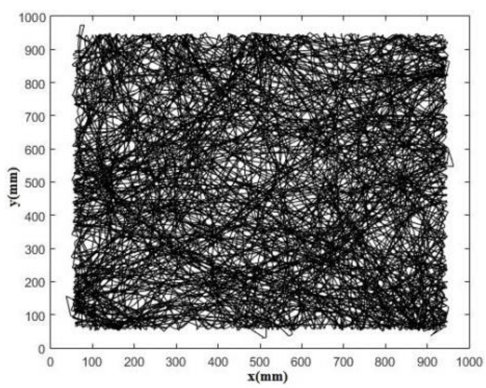

Figure 4. Simulation of chaotic trajectory for the robot in the x-y plane (environment a).

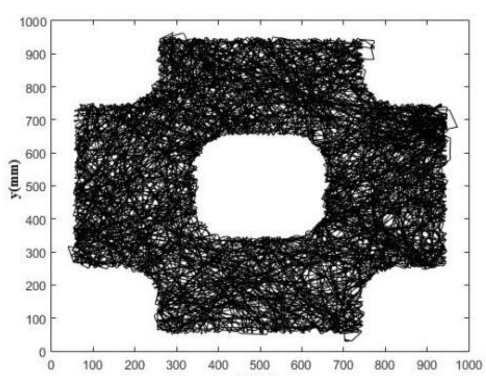

Figure 5. Simulation of chaotic trajectory for the robot in the x-y plane (environment b).

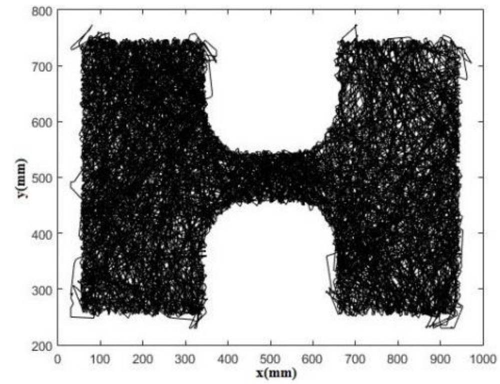

Figure 6. Simulation of chaotic trajectory for the robot in the x-y plane (environment c).

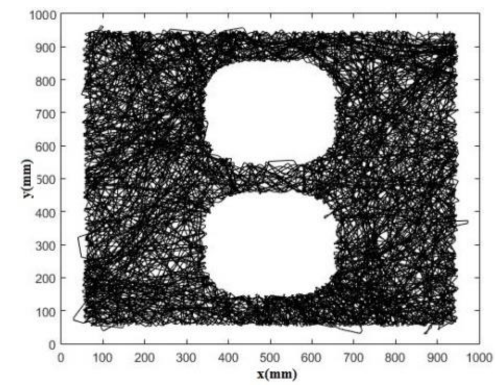

Figure 7. Simulation of chaotic trajectory for the robot in the x-y plane (environment d).

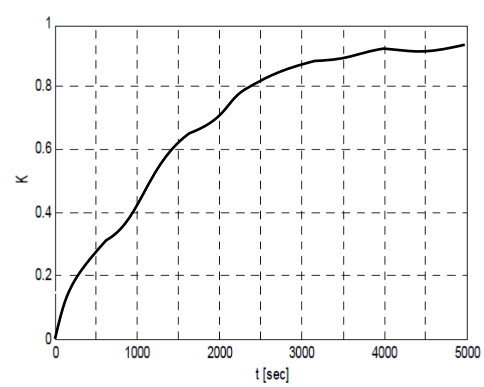

Figure 8. Performance index k for (environment a).

8. Proposed System Versus Arnold System

To prove the enhancement of area coverage process by the proposed system, a comparison with a well-known area coverage system proposed by Arnold [9] has been done, where the two systems were tested to cover an environment limited to (1000mm x 1000mm) and the designed robot operated using the conditions stated below:

Linear velocity: v = 0.15 m/s

Initial states: $[x_1\ x_2\ x_3]^T = [4\ 3.5\ 0]^T$

Work space limits: 1000 x 1000 mm

Simulation interval: 7000 sec

Figs. 9 and 10 show the samples of simulation result of the proposed system and Arnold chaotic system respectively. The results of simulation for the two methods shows that the rate of coverage area (k) in the case of the proposed method is greater than that of the Arnold method by about 10%, where the ratio (k) for the proposed and Arnold system was 96.62% and 84.15% respectively.

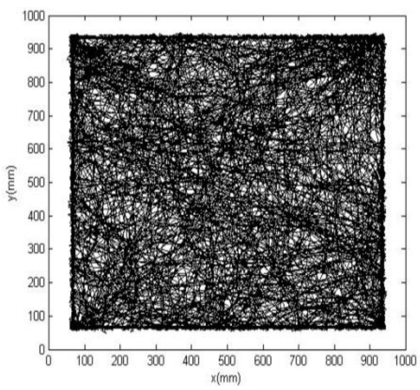

Figure 9. Simulation result for the robot based on the proposed system.

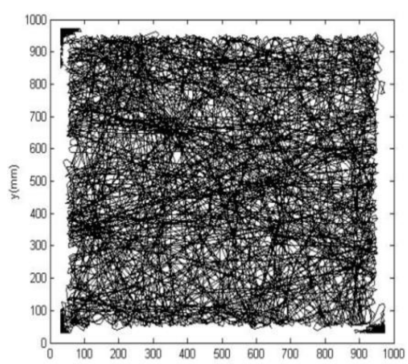

Figure 10. Simulation results of the robot based on the Arnold system.

Fig. 11 shows the rate of coverage area versus time in seconds for both chaotic systems and it is clear that the enhancement in the ratio of coverage area for the proposed system.

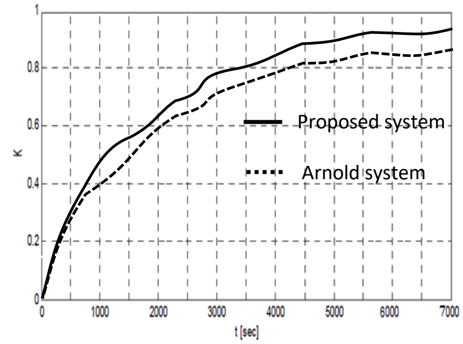

Figure 11. The rate of coverage area versus time in second for both chaotic systems.

9. Experimental Results

In this research the experimental data that related to the coverage performance of the designed chaotic mobile robot using the proposed chaotic motion controller, are presented. Experimental tests are necessary because that in real-world applications some dynamical factors such as internal and external friction and other workspace restrictions play a crucial role to mobile robot's behavior. Fig. 12 shows the workspace to be scanned by the designed mobile robot while Fig. 13 shows the autonomous mobile robot during the experimental test. Fig. 14 shows several waypoints during the navigation of the designed robot.

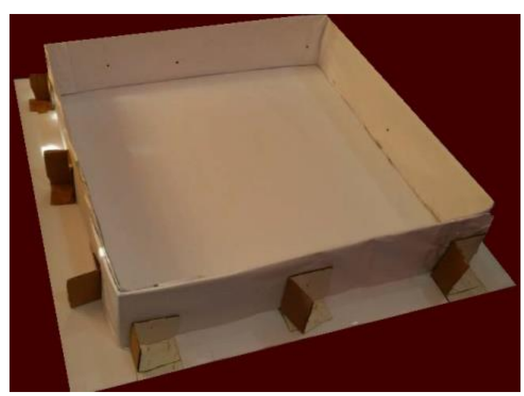

Figure 12. The workspace to be scanned by the designed mobile robot.

Figure 13. Terrain coverage using the proposed chaotic autonomous mobile robot.

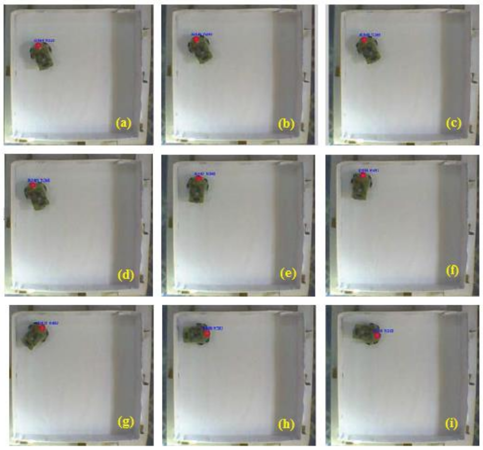

Figure 14. Several snapshots for the navigation of the designed robot.

Using a program developed with Matlab and a small web camera connected to the laptop, the robot trajectory was traced. This program was written to trace a small red ball attached to the mobile robot. Fig. 15 shows the trajectories acquired by the camera.

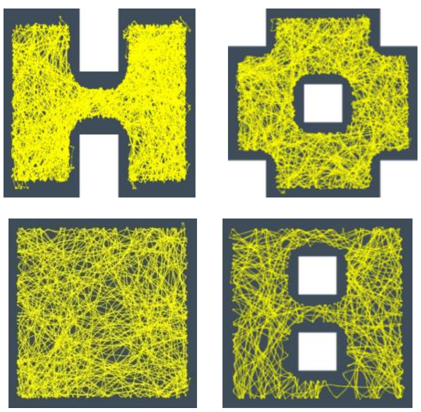

Figure 15. The trajectories acquired by the webcam.

10. Conclusions

Simulation tests for the new proposed chaotic system are performed using KiKS simulator program, where the new chaotic system is used in design a controller for autonomous robot. Chaotic mobile robot is implemented for scanning area without needing any map.

Four types of environment were designed to show the ability of the chaotic mobile robot from scanning all connecting workspaces. The performance index k, which represents the ratio of scanned area to the total area, is used to evaluate the efficiency of the autonomous mobile robot in covering the area. Finally, a comparison between Arnold chaotic system and the proposed new chaotic system was performed. The results confirm that the chaotic mobile robot was succeeded in covering all types of arenas in fast and complete manner with a very good average performance as compared with that of Arnold system.

11. References

1. Hanafi Dirman, Yousef Moh Abueejela, and Mohamad Fauzi Zakaria, (2013). *"Wall follower autonomous robot development applying fuzzy incremental controller"*, Intelligent Control and Automation, vol. 4, no. 1.
2. Martins Filho, Luiz S., et al. (2005). *"Kinematic control of mobile robots to produce chaotic trajectories"* Proc. of the 18th Int. Congress of Mechanical Engineering, Ouro Preto.
3. Siciliano B. and Khatib O. (Eds.), (2008). Springer Handbook of Robotics, Springer-Verlag Berlin Heidelberg, Germany.
4. Tarjoman, Mana, and Shaghayegh Zarei, (2009). *"The chaotic robot prediction by neuro fuzzy algorithm"* Conference of the international journal of arts and sciences. Vol. 1.
5. L. Kocarev, (2001). IEEE Circuits and Systems Magazine, vol. 1, no. 3, pp. 6–21.
6. Volos, Ch K., et al, (2013). *"Chaotic autonomous mobile robot for military missions."* Proceedings of the 17th

International Conference on Communications.

7. K. Cuomo, A. Oppenheim, and S. Strogatz, (1993). *“Synchronization of lorenz-based chaotic circuits with applications to communications,”* IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 40, no. 10.
8. Nilsson, Theodor, (2001*). "Kiks is a khepera simulator."* Stockholm, Sweden, Umea University.
9. Nakamura, Yoshihiko, and Akinori Sekiguchi, (2001). *"The chaotic mobile robot"* IEEE Transactions on Robotics and Automation. vol. 17, no. 6, pp. 898-904.