People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific research
M'hamed Bougara University, Boumerdes
Institute of Electrical and Electronic Engineering,
Laboratory of Signals and Systems (LSS)

ALGERIAN JOURNAL OF SIGNALS AND SYSTEMS

ISSN : 2543-3792

Title: **Control Of Mobile Robot Navigation Under The Virtual World Matlab-Gazebo**

Authors: **BENSACI Chaima [(1)], ZENNIR Youcef [(2)], POMORSKI Denis [(3)]**

Affiliations:
[(1)] **Université 20 Août 1955, LGCES Laboratory, Skikda, Algeria**
[(2)] **Université 20 Août 1955, Automatic Laboratory of Skikda, Skikda, Algeria**
[(3)] **Université Lille 1, CRIStAL Laboratory– UMR 9189, Lille, France**

Page range: **207- 217**

Volume : 2 Issue : 4 (December 2017)
A special issue of the International Conference on Advanced Engineering in Petrochemical Industry (ICAEPI'17)
November 28-30, 2017, Skikda, Algeria

Laboratory of Signals and Systems
Address : IGEE (Ex-INELEC), Boumerdes University, Avenue de l'indépendance, 35000, Boumerdes, Algeria
Phone/Fax : 024 79 57 66
Email : lss@univ-boumerdes.dz ; ajsyssig@gmail.com

Control Of Mobile Robot Navigation Under The Virtual World Matlab-Gazebo

BENSACI Chaima (1), ZENNIR Youcef (2)*, POMORSKI Denis (3)
(1) Université 20 Août 1955, LGCES Laboratory, Skikda, Algeria
(2) Université 20 Août 1955, Automatic Laboratory of Skikda, Skikda, Algeria
(3) Université Lille 1, CRIStAL Laboratory– UMR 9189, Lille, France
ch.bensaci@univ-skikda.dz; *y.zennir@univ-skikda.dz; denis.pomorski@univ-lille1.fr

Abstract: In this paper, we present our navigation control approach of a mobile robot (Turtlebot 2 robot) based on the stability Lyapunov function; our mobile robot is composed of two differential wheels. The kinematic model of the robot is presented followed by the description of the control approach. A 3D simulation under the Gazebo software is developed in interaction with the kinematic model and the control approach under MATLAB-SIMULINK software. The purpose of this study is to carry out an autonomous navigation; we initially planned different trajectories then we tried to be followed them by the robot. Our navigation strategy based on its odometry information, based on robot position and orientation errors; Velocity commands are sent for the robot to follow the chosen path. Different simulations were performed in 2D and 3D and the results obtained are presented followed by the envisaged future work.

Keywords: Autonomous navigation, Gazebo environment, Trajectory following, mobile robot (Turtlebot 2), kinematic model, lyapunov stability function, 2D and 3D simulation.

1. INTRODUCTION

The world of robotics is becoming bigger, dynamic and powerful. In our time there is a wide variety of robots with different types depending on the purpose of manufacture. The autonomous operation is very appreciable and desirable by the Man and the robots help to realize this operation. Robots are everywhere in our lives (house, hotel, at the bottom of the sea, in space, in industry). Mobile robots are particularly used in industry and homes with their ability to move with precision and stability to facilitate difficult tasks. As progress in this area is continuous and progressive, the main challenge of the industrial world is to take advantage of these advances and to integrate mobile robots to move products, especially in places that threaten the lives of workers such as chemistry laboratories. Autonomous robot navigation is necessary [1-3], in this context this study has been proposed. Different works in the literature have studied the navigation control of the mobiles robots with different control and navigation approaches [1-10]. In this article, we will study the planned navigation control of a mobile robot with differential wheels. We are going to control the Turtlebot 2 robot simulated under Gazebo world using Matlab and Simulink software. Speed commands are delivered to control the positions of the robot with respect to the path followed. The work is organized as follows: section 1 is an introduction, the autonomous mobiles robots navigation explained in section 2. In section 3, we present our mobile robot with its kinematic model. The control architecture is described in section 4. Finally, 2D and 3D simulation results are showed in section 5, 6 and a conclusion in section 7.

2. AUTONOMOUS MOBILE ROBOTS NAVIGATION

Autonomous systems are systems that develop, for themselves, the laws and strategies according to which they regulate their behavior: they are self-governing as well as self-regulating. Therefore, autonomous mobile robots navigation means that these robots have the ability to navigate automatically by their own selves without the intervention of human being whose role is limited to supervision [1] [2]. Autonomous navigation requires a number of heterogeneous capabilities, including the ability to execute elementary goal-achieving actions, like reaching a given location; to react in real time to unexpected events, like the sudden appearance of an obstacle; to build, use and maintain a map of the environment; to determine the robot's position with respect to this map; to form plans that pursue specific goals or avoid undesired situations; and to adapt to changes in the environment [3-11]. An autonomous navigation system consists of three main steps: Environment perception and localization, trajectory planning and robot control as shown in the following figure.

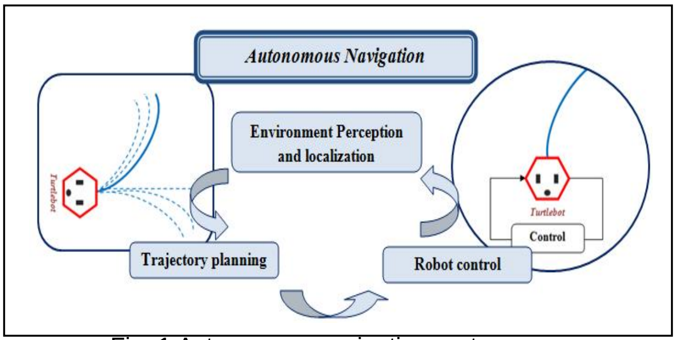

Fig. 1 Autonomous navigation system.

Navigation strategies

Diverse mobile robot navigation strategies are adopted to allow mobile robot to move, avoid obstacles and reach the goal, Trullier et al have been classified them into five categories [4].

Approach of an object

This basic capacity makes it possible to move towards a visible object from the current position of the robot. It is generally performed by a gradient rise based on the perception of the object. This strategy uses reflex actions, in which each perception is directly associated with an action.

Guidance

This ability seems to be used by certain insects; it allows to reaches a goal that is not a directly visible material object but a point of space characterized by the spatial configuration of a set of remarkable objects. This strategy also uses reflex actions.

Action associated with a location

This strategy makes it possible to reach a goal from positions for which this goal or the landmarks that characterize its location are invisible. It requires an internal representation of the environment which consists of defining places as zones of space in which perceptions remain similar, and associating an action to be performed at each of these places. The sequence of actions defines a road that allows reaching the goal.

Topological navigation

It is an extension of the previous strategy which memorizes in the internal model the spatial relations between the different places. These relationships indicate the possibility of moving from one place to another. Therefore the internal model is a graph that makes it possible to calculate different paths between two arbitrary places. This model, however, allows only the planning ofdisplacements among the known places and following the known paths.

Metric navigation

This strategy allows the robot to plan paths within unexplored areas of its environment. It memorizes for this the relative metric positions of the different places, in addition to the possibility of passing from one to the other. By simple composition of vectors these relative positions can calculate a trajectory, going from one place to another, even if the possibility of this displacement has not been memorized in the link form.

Systems and Methods for mobile robot navigation

Odometry and Other Dead-Reckoning Methods

Odometry is one of the most commonly adopted navigation methods for mobile robot positioning.This method uses encoders to measure wheel rotation and/or steering orientation. It provides good short-term accuracy, an inexpensive facility, and allows high sampling rates.

Inertial Navigation

This method uses gyroscopes and sometimes accelerometers to measure rate of rotation and acceleration. Measurements are integrated once (or twice) to yield position. Inertial sensors are thus unsuitable for accurate positioning over an extended period of time.

Active Beacon Navigation Systems

This method computes the absolute position of the robot from measuring the direction of incidence of three or more actively transmitted beacons. The transmitters, usually using light or radio frequencies must be located at known sites in the environment.

Landmark Navigation

In this method distinctive artificial landmarks are placed at known locations in the environment. The advantage of artificial landmarks is that they can be designed for optimal detectability even under adverse environmental conditions. but this approach is computationally intensive and not very accurate.

Map-based Positioning

In this method information acquired from the robot's onboard sensors is compared to a map or world model of the environment. If features from the sensor-based map and the world model map match, then the vehicle's absolute location can be estimated. Map-based positioning often includes improving global maps based on the new sensory observations in a dynamic environment and integrating local maps into the global map to cover previously unexplored areas. The maps used in navigation include two major types: Geometric maps and topological maps. Geometric maps represent the world in a global coordinate system, while topological maps represent the world as a network of nodes and arcs [3].

Reactive Navigation

Reactive navigation differs from planned navigation in that, while a mission is assigned or a goal location is known, the robot does not plan its path but rather navigates itself by reacting to its immediate environment in real time. Reactive methods share a number of important features:

- First, sensors are tightly coupled to actuators through fairly simple computational mechanisms.
- Second, complexity is managed by decomposing the problem according to tasks rather than functions [3].

In this paper we use the Turtlebot 2 robot simulated in gazebo to make a planned autonomous navigation; we represent as a first step different trajectories and try to follow them.

3. TURTLEBOT 2 ROBOT DESCRIPTION

The is one of the non-holonomic robots officially proposed by Willow Garage to develop in the operating system dedicated to robotics: ROS. It is equipped with a Kinect sensor, a Netbook, trays for the installation of these two components and a Kobuki base as shown in figure 2.

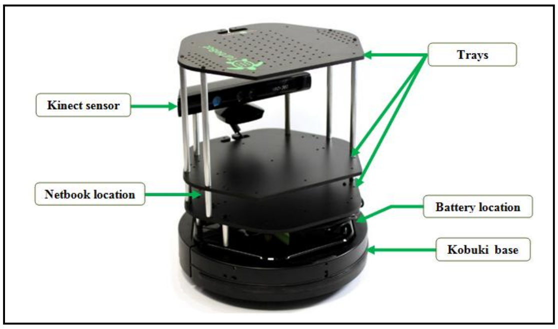

Fig. 2 Components of turtlebot 2.

The Kobuki base is a mobile wheeled robot with two differential wheels and a castor wheel, it also contains proximity sensors, encoders on the wheels and a gyrometer for each axis [5].The Turtlebot 2 is moved by the following model: We consider that (X, Y) is the global coordinate system and (XR, YR) is the local coordinate system of the robot. The position of the robot P (x, y, θ) is represented in Cartesian coordinates in the global coordinate system.

The relationship between the robot frame and the global frame is given by the basic transformation matrix:

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The figure below show the relationship between the two frames.

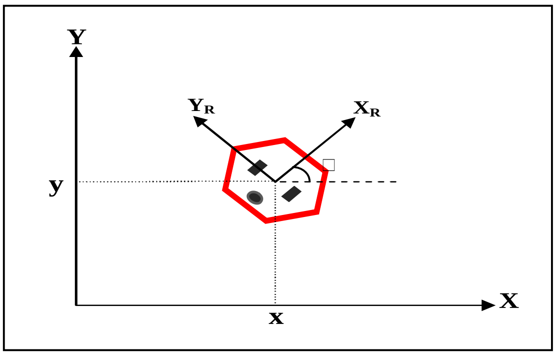

Fig. 3 Turtlebot 2 representation in a Cartesian coordinate system.

The wheels are motorized independently. When both wheels turn with the same speed in the direct action the robot moves forward otherwise it moves backwards. Turning right is done by actuating the left wheel at a higher speed than that of the right wheel and vice versa to turn left. It also can rotate on the spot by actuating a wheel forward and the second wheel in the opposite direction with the same speed. The third wheel is a free wheel preserve the robot stability. The Turtlebot 2 wheels are non-holonomic, they represent non holonomic constraints:

$$\dot{x}\sin\theta - \dot{y}cos\theta = 0 \quad (2)$$

The relationship between the speeds of the robot (v, w) and the speeds of the wheels (wr, wl) is expressed by this two equations:

$$v = v_x = v_1 + v_2 = r(\frac{w_r+w_l}{2}) \quad (3)$$

$$w = w_1 + w_2 = r(\frac{w_r-w_l}{d}) \quad (4)$$

Where r is the wheel radius and d is the distance between wheels.

Kinematic equations

Suppose that the robot is in an arbitrary position P (xc, yc, θ) and the distance between its current position and the desired position R (xd, yd, β) ; defined with reference to the global frame; higher than 0.

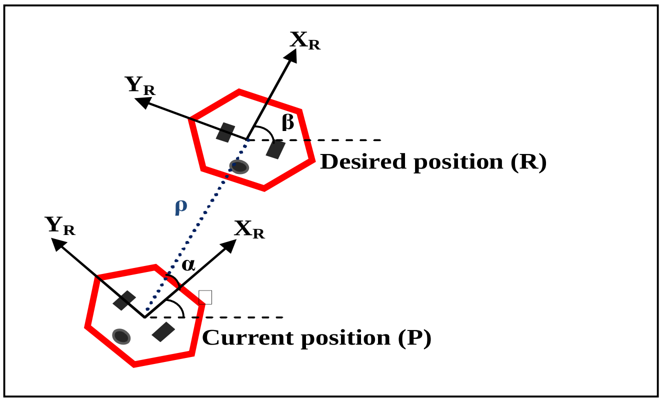

Fig. 4 Turtlebot 2 position and orientation error vectors.

The robot Cartesian is given by [11-13]:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = w \end{cases} \quad (5)$$

Where: x, y and θ are measured relative to the global frame.
The robot position can also be represented by the polar coordinates with a distance error ρ>0:

$$\begin{cases} \dot{\rho} = -v\cos(\beta-\theta) = -v\cos\alpha \\ \dot{\beta} = v\frac{\sin\alpha}{\rho} \\ \dot{\theta} = w \end{cases} \quad (6)$$

Where $\alpha = \beta - \theta$ is the angle measured between the axis X_R of the reference relative to the robot and the axis ρ expressed by the following equation:

$$\rho = \sqrt{(xd - xc)^2 + (yd - yc)^2} \quad (7)$$

The kinematic equations obtained on the basis of the polar coordinates are:

$$\begin{cases} \dot{\rho} = -v\cos\alpha \\ \dot{\alpha} = -w + v\frac{\sin\alpha}{\rho} \\ \dot{\beta} = v\frac{\sin\alpha}{\rho} \end{cases} \quad (8)$$

This set of equations is only valid for $\rho \neq 0$.

Control law and stability

The control algorithm must be designed to move the robot from its current configuration to its desired configuration, so we are going to send linear and angular velocity commands $u = \binom{v}{w}$ until the robot reaches the desired position.The proposed control law is state-dependent ie. $\binom{v}{w} = f(\rho, \alpha, \beta)$, which ensured that the state is moving toward (0, 0, β) without reaching $\rho = 0$ in a finite time interval. One of the most commonly used methods for studying asymptotic behavior is based on Lyapunov stability theory. Consider a simple positive definite quadratic form of the Lyapunov function:

$$V = V_1 + V_2 = \frac{1}{2}\rho^2 + \frac{1}{2}\alpha^2 \quad (9)$$

ρ *et* α are respectively the distance and orientation errors.
Its derivative relative to time is given by:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = \dot{\rho}\rho - \dot{\alpha}\alpha \quad (10)$$

Using the kinematic equations:

$$\dot{V} = (-v\cos\alpha) + \alpha\,(-w + v\,\frac{\sin\alpha}{\rho}) \tag{11}$$

The first term can be non-positive, putting the linear velocity in the form:

$$\begin{cases} v = K_\rho\,\rho\cos\alpha \ \text{ avec } K_\rho > 0 \\ \dot{V}_1 = \rho\,(-K_\rho\ \rho\cos^2\alpha) \\ \dot{V}_1 = -K_\rho\rho^2\cos^2\alpha \le 0 \end{cases} \tag{12}$$

This means that the term $\dot{V}_1$ is always non-increasing over time, therefore it converges asymptotically to a non-negative finite limit. Likewise, the second term can be non-positive thus the angular velocity is put in the form:

$$w = K_\rho\sin\alpha\cos\alpha + K\alpha \quad \text{With } K\alpha > 0 \tag{13}$$

$$\dot{V}_2 = \alpha\,(-K_\rho\sin\alpha\cos\alpha - K\alpha\,\alpha - \frac{K_\rho\,\rho\,\sin\alpha\cos\alpha}{\rho}) \tag{14}$$

$$\dot{V}_2 = -K\alpha\,\alpha^2 \le 0 \tag{15}$$

Finally, the expression of the time derivative of the Lyapunov function V becomes:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 = -K_\rho\,\rho^2\cos^2\alpha - K\alpha\,\alpha^2 \le 0 \tag{16}$$

The result is in semi-negative form. By applying the Barbalat lemma, it follows that V□ necessarily converges to zero on time. Which implies that the state vector convergence from (ρ, α, β) to (0, 0, β).So we conclude that the expressions of the following linear and angular velocities make the movement of the robot smooth and stable [5].

$$\begin{cases} v = K_\rho\,\rho\cos\alpha \\ w = K_\rho\sin\alpha\cos\alpha + K\alpha\,\alpha \end{cases} \tag{17}$$

4. CONTROL ARCHITECTURE PROPOSED

The planned motion of the turtlebot 2 robot must conform to the chosen trajectory. Therefore, we have chosen a control law based on the regulation of linear and angular velocities depending to the distance and the orientation errors between the desired positions and the current positions of the robot. The nice tracking of these trajectories is a first step on the success way of the desired autonomous navigation.The figure below represent the the proposed control architecture loop [11-13].

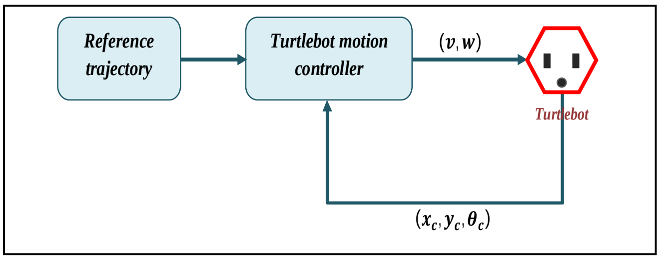

Fig. 5 The proposed control architecture loop.

The desired trajectories equations

The reference trajectory can give the destination according to the current position and the trajectory. The controller is tested on different reference trajectories:

Chosen trajectories equations defined as follows :

- *Infinite trajectory :*

$$\begin{cases} x = 9.(\cos(2t) - 1).\cos(t) \\ y = 9.(\cos(2t) - 1).\sin(t) \end{cases} \tag{18}$$

$$\begin{cases} x = 9.(\cos(2t) + 1).\cos(t) \\ y = 9.(\cos(2t) + 1).\sin(t) \end{cases} \tag{19}$$

- *Circle trajectory :*

$$\begin{cases} x = 18.\cos(t) \\ y = 18.\sin(t) \end{cases} \tag{20}$$

- *Ellipse trajectory* :

$$\begin{cases} x = t \\ y = x^2 - 19 \end{cases} \tag{21}$$

$$\begin{cases} x = 18.(\cos(2t) - 1).\cos(t) \\ y = 18.(\cos(-2t) + 1).\sin(t) \end{cases} \tag{22}$$

- *Inclined straight* :

$$\begin{cases} x = 20.\cos(t).\sin(-t) \\ y = 20.\sin(\mathrm{t}).\cos(-t) \end{cases} \tag{23}$$

5. SIMULATION UNDER MATLAB

We have used this strategy of control with reference trajectory of infinity trajectory. Different trajectories are used also as follow:

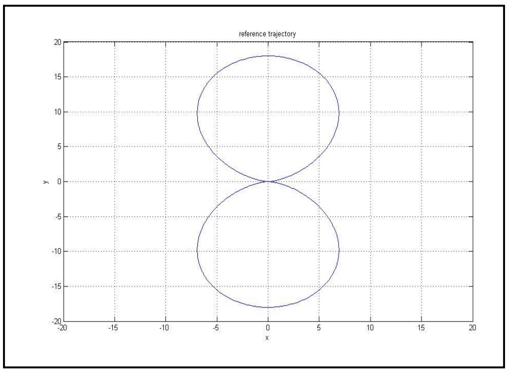

Fig. 6 Reference infinite trajectory .

Infinite trajectory

- *Trajectory following*

The following figure show the Turtlebot 2 follows the desired trajectory:

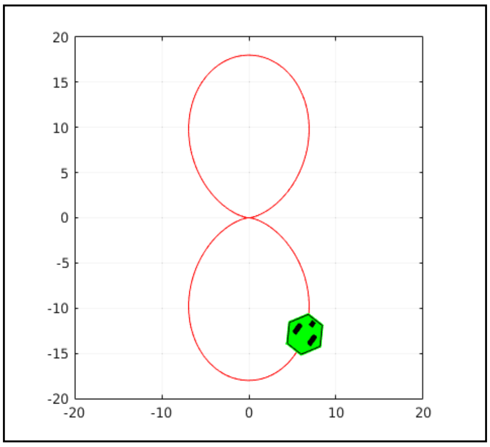

Fig. 7 Turtlebot 2 follow the infinite trajectory 1 .

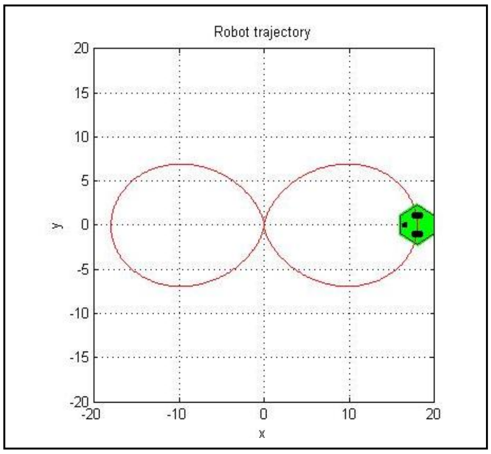

Fig. 8 Turtlebot 2 follow the infinite trajectory 2.

- *Trajectory tracing*

The following figure show the turtlebot trace the trajectory during its navigation :

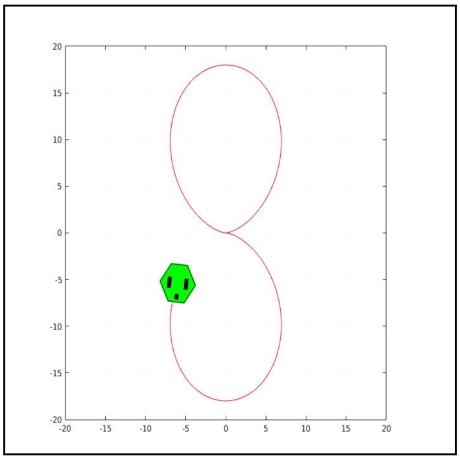

Fig. 9 Infinite trajectory tracing by the movement of Turtlebot 2.

Circle and Ellipse trajectories

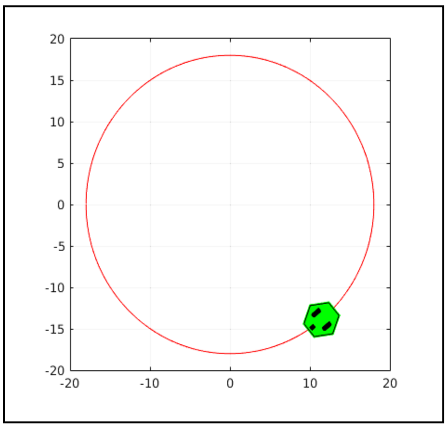

Fig. 9 Circle trajectory following by Turtlebot 2.

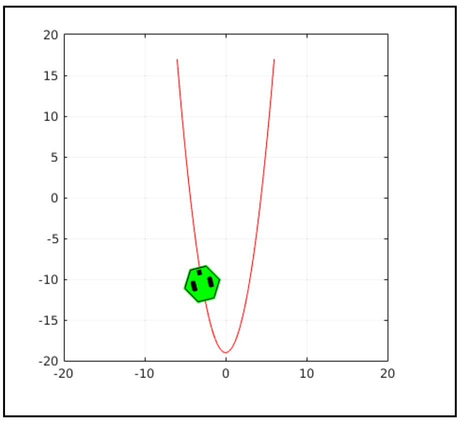

Fig. 10 Ellipse trajectory following by Turtlebot 2.

Butterfly and Inclined straight trajectories

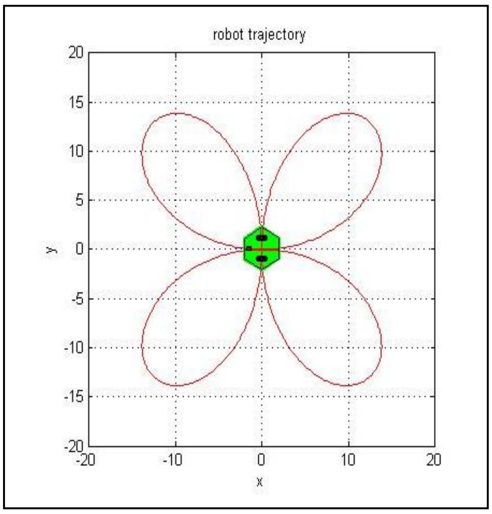

Fig. 11 Butterfly trajectory following by Turtlebot 2.

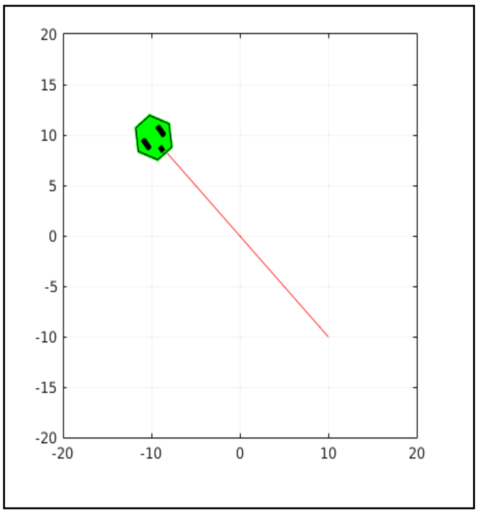

Fig. 12 Inclined straight trajectory tracing by the movement of Turtlebot 2.

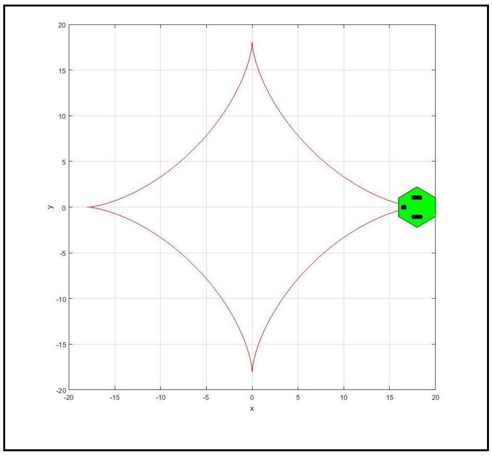

Fig. 13 Trajectory following by Turtlebot 2.

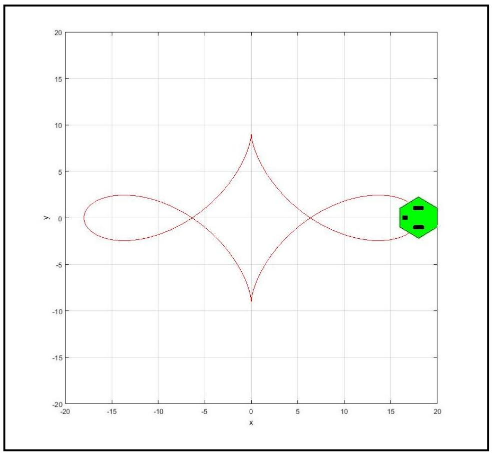

Fig. 14 Trajectory following by Turtlebot 2.

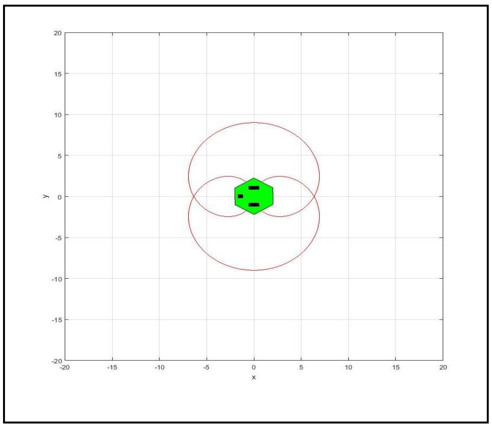

Fig. 13 Trajectory following by Turtlebot 2.

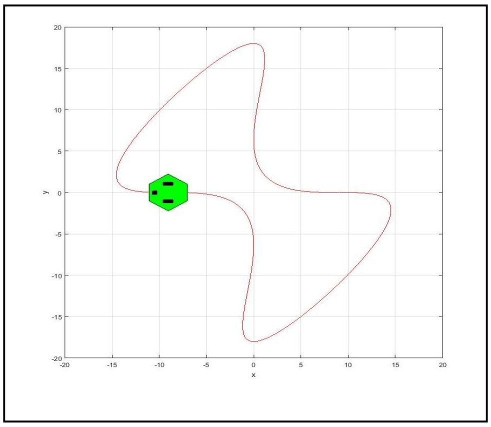

Fig. 14 Trajectory following by Turtlebot 2.

6. 3D SIMULATION UNDER GAZEBO

Gazebo is a robotic three-dimensional simulator for indoor and outdoor environments. it is capable of simulating a population of robots, sensors and objects. It generates both realistic sensor feedback and physically plausible interactions between objects (it includes an accurate simulation of rigid body physics).By realistically simulating robots and environments code designed to operate a physical robot can be executed on an artificial version. Numerous researchers have also used Gazebo to develop and run experiments solely in a simulated environment. Gazebo simulation can be regarded as a experimental copy of real robot in virtual world [11-15].In our work we use the simulated turtlebot2of gazebo world, the figure below show the turtlebot 2 simulated robot under gazebo world.

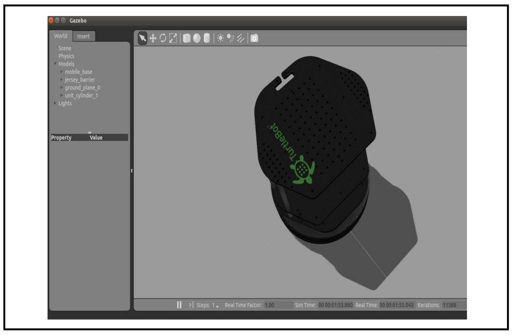

Fig. 15 Turtlebot 2 robot simulated under gazebo world.

The result of 3 D simulation with turtlebot2 in gazebo world when using an infinite reference trajectory is represented in the following figure:

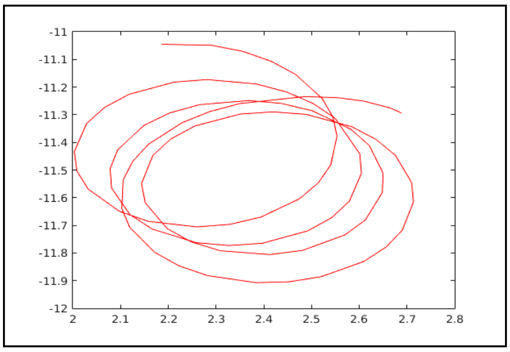

Fig. 16 Output with Turtlebot 2 in gazebo world .

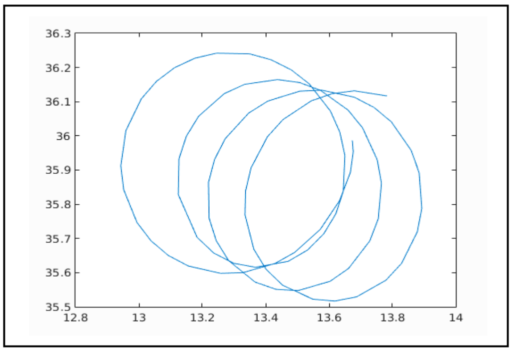

Fig. 17 Output with Turtlebot 2 in gazebo world .

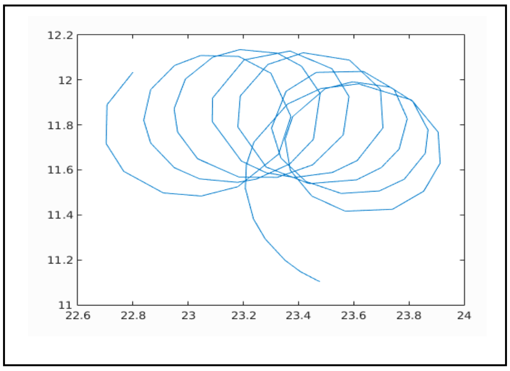

Fig. 18 Output with Turtlebot 2 in gazebo world .

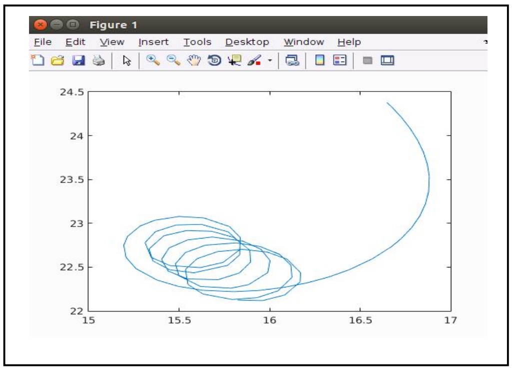

Fig. 19 Output with Turtlebot 2 in gazebo world .

We note from this result that the robot does not follow the desired trajectory. In the future work we will try to solve this issue by improving the robot control.

7. CONCLUSION

We have presented in this paper the functioning principle of our robot (Turtlebot 2) with its kinematic model used for the navigation. We have described the control strategy used to regulate the robot to track different reference trajectories. 3D simulation was developed in Gazebo world using Matlab-Simulink. The obtained results (fig.7 - fig.19) show the efficiency of navigation control strategy in Matlab simulation but with Gazebo we need more manipulation of this software and the control parameters. In the future work we will generalize this study with two or more robots with the use of artificial intelligence techniques for navigation control.

References

[1] Gilles Tagne Fokam. Commande et planification de trajectoires pour la navigation de véhicules autonomes. Autre. Université de Technologie de Compiègne, 2014.

[2] M. Spong, , S. Hutchinson, and M. Vidyasagar. Robot Modeling and Control, John Wiley & Sons, INC, New York, 2006,p.496.

[3] Luc Steels**,** When are robots intelligent autonomous agents? Robotics and Autonomous Systems 15, pp. 3-9,1995.

[4] Jin Cao, Vision Techniques and Autonomous Navigation for an Unmanned Mobile Robot, A.de Ruijter, Master thesis, the University of Cincinnati, Department of Mechanical, Industrial and Nuclear Engineering Of the College of Engineering,Tianjin University, China, 1995.

[5] David Filliat. Robotique Mobile. Engineering school. Robotique Mobile, ENSTA ParisTech, pp.175, 2011.

[6] R. M. DeSantis (1995). Modeling and path-tracking control of a mobile wheeled robot with a differential drive. Robotica, volume 13, pp 401-410 doi:10.1017/S026357470001883X

[7] M. Sampei, T. Tamura, T . Itoh and M. Nakamichi , Path Tracking Control of Trailer-Like Mobile Robot, IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91, Nov. 3-5. 1991,Osaka, Japan.

[8] Gianluca Antonelli et al, A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking, IEEE Transactions on fuzzy systems, Vol. 15, No. 2, April 2007

[9] B. Lakehal, Y. Amirat, and J. Pontnau, "Fuzzy steering control of a mobile robot," in Proc. IEEE Int. Conf. Indust. Autom. Contr.: Emerging Technol., Taipei, Taiwan, 1995, pp. 383–386.

[10] Takanori Fukao, Hiroshi Nakagawa, and Norihiko Adachi, Adaptive Tracking Control of a Nonholonomic Mobile Robot, IEEE Transactions on robotics and automation, Vol. 16, No. 5, October 2000.

[11] K. C. KOH AND H. S. CHO, A Smooth Path Tracking Algorithm for Wheeled Mobile Robots with Dynamic Constraints, journal of Intelligent and Robotic Systems 24: 367–385, 1999.

[12] R. Fierro and F. L. Lewis, Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics, Journal of Robotic Systems 14(3), 149–163 (1997)

[13] Yutaka Kanayama et al, A Stable Tracking Control Method for an Autonomous Mobile Robot, Calhoun: The NPS Institutional Archive, 1990.

[14] F.WEHBI, "Architecture logicielle et matérielle pour fusion tolérantes au fautes pour systèmes multi-robots sous ROS". Rapport, Lille and Libanon University, pp.64, 2015.

[15] Wei Qian, Zeyang Xia, Jing Xiong, Yangzhou Gan, Yangchao Guo, Shaokui Weng, Hao Deng, Ying Hu, Jianwei Zhang, Manipulation Task Simulation using ROS and Gazebo, Proceedings of the 2014 IEEE International Conference on Robotics and Biomimetics December 5-10, Bali, Indonesia, 2014.