A NOVEL NAVIGATION STRATEGY FOR AN UNICYCLE MOBILE ROBOT INSPIRED FROM THE DUCK WALKING (GYROSCOPIC NAVIGATION)

Radwen Bahri[1], Rahma Boucetta[2] And Saloua Bel Hadj Ali Naoui[3]

[1]University Of Gabes, National Engineering School Of Gabes, Tunisia
Research Lab. Modeling, Analysis And Control Of Systems (MACS) LR16ES22
[2]Sciences Faculty of Sciences, University of Sfax, Soukra Avenue, 3000 Sfax Tunisia
[3]Preparatory Institute for Engineering Studies, El Manar University, Tunis , Tunisia

ABSTRACT

This paper deals with the design and the implementation of a novel navigation strategy for an unicycle mobile robot. The strategy of navigation is inspired from nature (the duck walking) in the closed loop. Therefore, a single sensor (gyroscope) is used to determine the pose of the robot, hence the proposed name "gyroscopic navigation". A global presentation of the novel strategy and the robot and with its different components are given. An experimental identification for the useful parameters are then exhibited. Moreover, the controller design strategy and the simulation results are given. Finally, real time experiments are accomplished to valid the simulation results.

KEYWORDS

Robot navigation, navigation strategy, Robot pose, angular position

1. INTRODUCTION

Nowadays, a great variety of navigation strategies have been developed. Nevertheless, these strategies are limited to odometry in order to determine the position and the speed of the robot. In this regard, the simplest way to measure displacement and speed of a mobile robot is to place an encoder in the drive motors in the active wheels [1, 2, 3, 4] or even in additional passive wheels [5]. However, measurements given by the encoder suffer from many problems such as accurate wheel diameter estimation, wheel wear and slippage and uncertainty in the determination of the contact point with the floor. Actually, there are various techniques for measuring the position rather than odometry method. Doppler navigation systems use the doppler effect to determine the speed of vehicle. The principle of the process is based on the doppler shift in frequency observed when radiated energy reflects a surface that is moving with respect to the emitter. Those systems cause errors in detecting true ground, speed and temperature. They also affect the velocity of sound and oscillation frequency of a crystal oscillator which impact the velocity given by the sensor [6] and require the existence of a reflection surface. Furthermore, the global positioning systems gps suffer from signal masking in areas such as densely treed streets and tunnels [7][10] as well as sampling rate [9].

A novel navigation strategy is proposed to overcome these problems. This paper is arranged as shown. Section II presents the structural design and the strategy principle of the robot navigation. Section III presents mathematical model, simulation results, practical validations and specifications for the novel strategy. Finally, the paper is ended with a conclusion.

2. THE ROBOT DESIGN AND ARCHITECTURE

2.1. Robot design

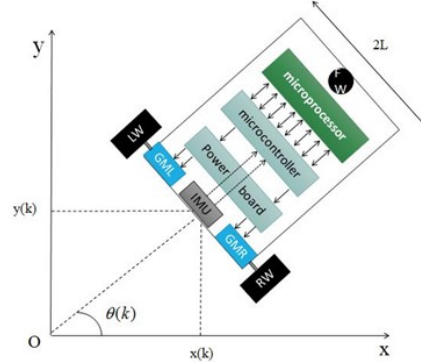The robot used in this study is an unicycle robot as shown in the following figure 1.

Figure 1: Robot architecture.

2.2. Robot architecture

The Robot comprises a microprocessor to implement a neural network for the path planning which requires high calculation capacity and a microcontroller for interface with IMU (inertial measurement unit) and the power board managing the energy required for the operation of the actuators.

3. NAVIGATION STRATEGY

This section describes the origin of idea from the duck walk to develop a functional analysis and to show a mathematical description of the evolution of the robot in the plan.

3.1. Functional Analysis

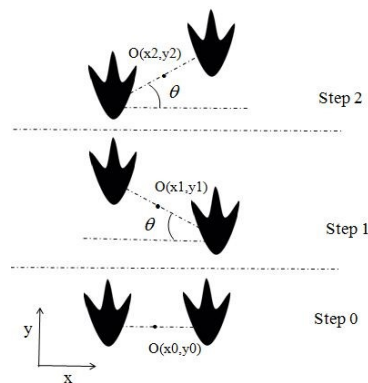An example of a duck walk is given in the figure 2.

Figure 2: The duck walk

Where O is the center of gravity and (x,y) are the Cartesian coordinates in the plane. It is assumed that the evolution medium of duck is a non-inclined plane, the direction is constant and we are interested in the evolution of the Y-coordinate. So, we obtain:

$$y(k+1) = y(k) + \Delta y \qquad (1)$$

where

$$\Delta y = l.\sin(\Delta(\theta)) \qquad (2)$$

and

$$\theta(k+1) = \theta(k) + \Delta(\theta) \qquad (3)$$

The same model is given on the x-axis

$$x(k+1) = x(k) + \Delta x \qquad (4)$$

where

$$\Delta x = l.\sin(\Delta(\theta - \pi/2)) \qquad (5)$$

3.2 Path Planning

It is assumed that the evolution medium of the duck is a non-inclined plane, xf > x0 and yf > y0. The new strategy relies on the change of the angular position of the driving wheels, one of which plays the role of the center of rotation and the other wheel moves.

The robot posture in base frame is given by $q(k) = [x(k) \; y(k) \; \theta(k)]^T$.

We assume that the initial pose of the robot is $q_i = [x_i \; y_i \; \theta_i]^T$ and the final pose is $q_f = [x_f \; y_f \; \theta_f]^T$. The trajectory is divided into two parts: the first one is a straight line parallel to the axis Y [Dy] with

$$Dy = [y_i \; y_f] - L \qquad (6)$$

Then Dy is divided to n steps Y_s with

$$Dy = n.Y_s \qquad (7)$$

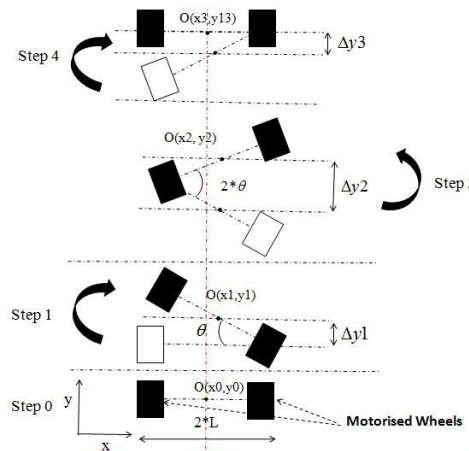Y_s is composed of 3 sub-steps as shown in Fig.3

Figure 3: Y evolution

We pose

$$Y_s = \Delta y1 + \Delta y2 + \Delta y3 \qquad (8)$$

where

$$\Delta y1 = L.\sin(\theta) \tag{9}$$

$$\Delta y2 = 2L.\sin(\theta) \tag{10}$$

$$\Delta y3 = L.\sin(\theta) \tag{11}$$

The second part of the trajectory is a straight line parallel to the axis X [Dx] with :

$$Dx = [x_i \ x_f] - L \tag{12}$$

Then Dx is divided to n steps X_s with :

$$Dx = n.X_s \tag{13}$$

X_s is composed of 3 sub-steps as shown in Fig.4. We pose
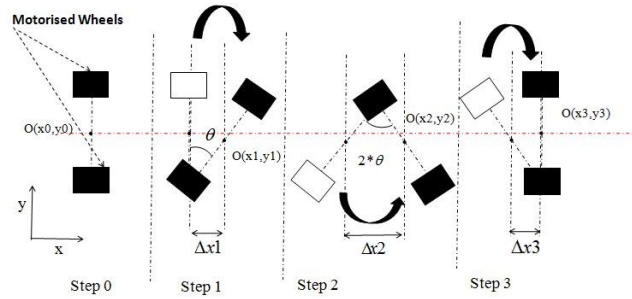
$$X_s = \Delta x1 + \Delta x2 + \Delta x3 \ 0 \tag{14}$$

Figure 4: X evolution

where

$$\Delta x1 = L.\sin(\theta - \Pi / 2) \tag{15}$$

$$\Delta x2 = L.\sin(2\theta - \Pi / 2) \tag{16}$$

$$\Delta x3 = L.\sin(\theta - \Pi / 2) \tag{17}$$

The generation of the trajectory is not a simple task, a classic regulator is insufficient, so it was thought to synthesize a mixed regulator based on artificial intelligence on the one part and the conventional method regulation on the other part.

3.3 Controller Design

In this work, we try to get rid of the rotary encoders to estimate the position of the robot in its trajectory by proposing a new navigation strategy based on a single sensor (gyroscope) which gives us the angular position. The calculation of the position which the robot must reached necessitates a special regulator by proposing the following regulators shown in the following fig. 5.
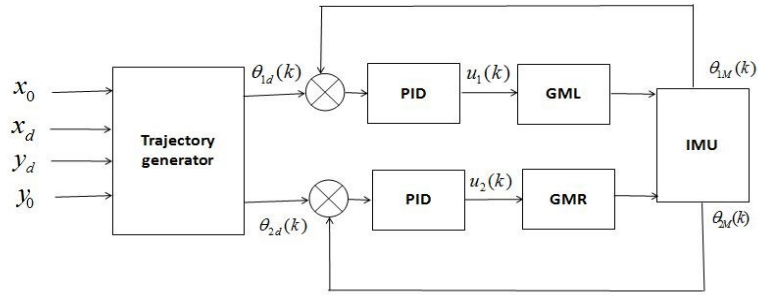
Figure 5: Controller Design

The neural network is chosen as a Multilayer Perceptron shown in fig. 6. The design, study and simulation of the neural network used as a trajectory generator (generate (θ)) in this work will be presented in the next subsection.

3.4 General Algorithm

The general algorithm of this strategy is divided in 5 steps:

- determine the distance $Dy = [y_i \ y_f] - L$
- determine the distance $Dx = [x_i \ x_f] - L$
- determine the number of step $Dy = n.Y_s$
- determine the number of step $Dx = n.X_s$
- move Dy
- turn $-\pi / 2$
- move Dx

3.5 Development of the robot

The robot was developed in our laboratory. It is composed of 2 motorized wheels ensuring its mobility (1, 2). An IMU (mpu9150 (7)) is introduced to determine the latitude of the robot (roll, pitch, and yaw) in the field of evolution, in our work we are interested only in the Yaw angle. There is also a raspberry pi 3 board that serves as a microcomputer (3) with hight performance (CPU 4 * ARM Cortex-A53, 1.2ghz with 1GB of RAM), its role is to generate trajectory to the robot by generating the desired angle θ. The robot embedded a microcontroler (5) to treats data coming from sensor and generates control signals to the electronic board (the power board is based on the integrated circuit L298N designed for the control of speed and direction of the DC motors) (6) which controlled the actuators. As power using a lipo battery 2s for the actuators and a power bank (5 volt 1000 mah) for the raspberry pi, the robot architecture is given by Fig.6.
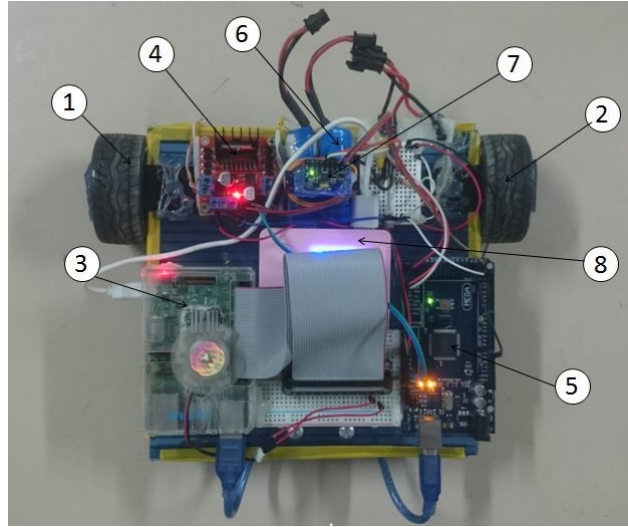
Figure 6: The developed prototype robot

The mass of the robot is 0.9 Kg grams with the dimensions $200 \times 210 \times 75 mm$ and the possibility of wireless communication with the PC.

4. MODELLING AND SIMULATION

4.0.1 Neural network learning and simulation

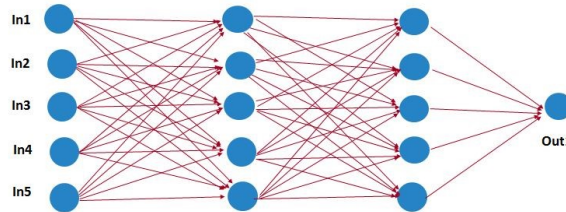In order to obtain a good trajectory generator, the neural network must be learned to ensure the desired behavior.

Figure 7: Multilayer Perceptron

In our case, the Inputs of the Neural Network are $In1 = xd$, $In2 = yd$, $In3 = x_0$, $In4 = y_0$, $In5 = t$ and the Outputs $Out1 = \theta_d$

4.0.2 Experimental identification

In order to control the robot, a mathematical model is required for the actuators. Experimental identification is used to determine the angular speed in terms of voltage, a step has been applied to the actuators shown in fig.8.
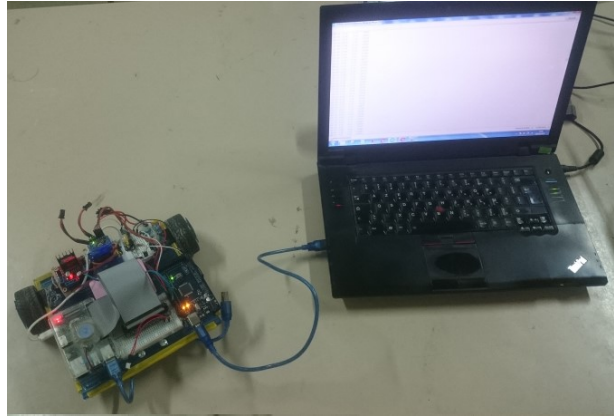
Figure 8: Experimental identification

The obtained output signal is assumed by a first-order system with a transfer function given by:

$$H(p) = \frac{K}{1+\tau p} \qquad (18)$$
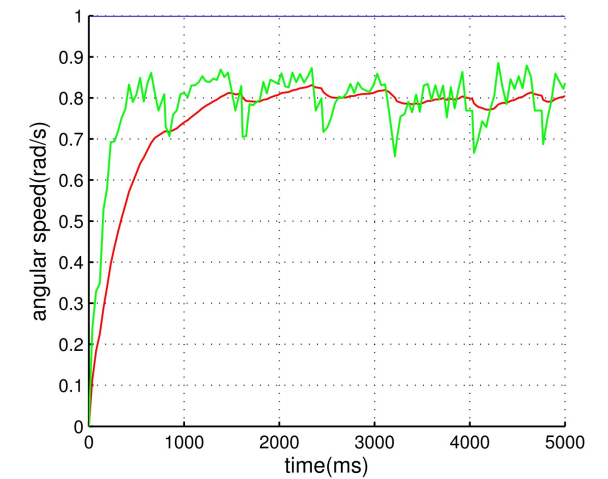
Figure 9: Step response

$K = 1$ And $\tau = 10 * Ts$. The system parameters can be determined graphically using a simple process, thereby give the following values $K = 0.8$ and $\tau = 0.24s$. After obtaining the mathematical model of the angular velocity, the model of the angular position can be obtained by simple integration. As a controller, a pid controller is chosen with three degrees-of freedom (k_p, T_i, T_d).

$$\begin{cases} k_p = 13 \\ T_i = 0.33 \\ T_d = 0.33 \end{cases}$$

The simulation of X_s and Y_s is given by the following figure.
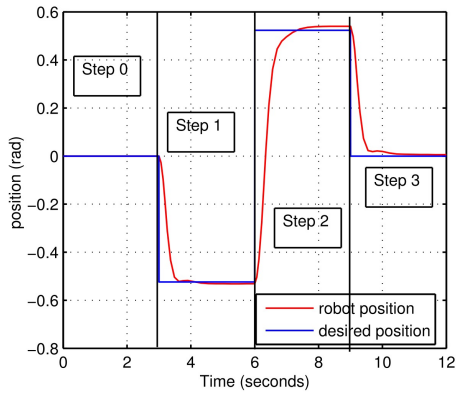
Figure 10: The simulation of Y_s
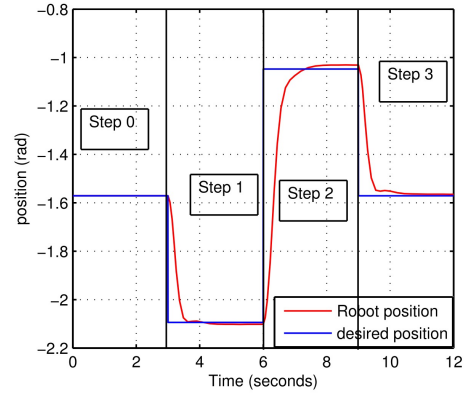
Figure 11: The simulation of X_s

where

      Step 0 is the intitial postion of the robot

      Step 1 present $\Delta y1$ respectively $\Delta x1$

      Step 2 present $\Delta y2$ respectively $\Delta x2$

      Step 3 present $\Delta y3$ respectively $\Delta x3$

The PID controller shows an adequate response with an acceptable overshoot and a zero steady-state error. The control signal maintains acceptable values by the actuators. In this simulation, the starting point of the initial robot posture in base frame is given by $q_i = [0\,0\,0]^T$ and the final pose is $q_f = [0.5\,0.5 - \pi/2]^T$

where

      $Dy = 2 * Ys$

      $Dx = 2 * Xs$
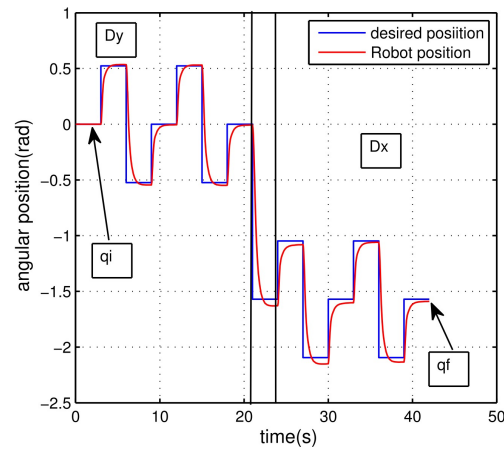
Figure 13: Planification trajectory simulation
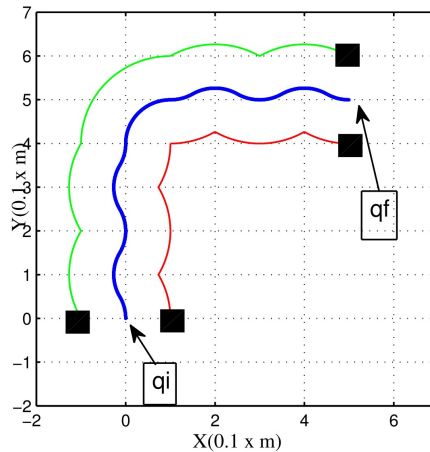
Figure 14: Trajectory simulation

Figures 14 and 15 present the simulation of the angular position and the total trajectory of the robot.

5. PRACTICAL VALIDATION

5.1 Implementation of the PID controller

To implement the PID controller in the real closed-loop system, the transfer function of the command signal must be discretized to deduce the digital control law as:

$$u(k) = u(k-1) + r0.\varepsilon(k) + r1.\varepsilon(k-1) + r2.\varepsilon(k-2) \qquad (19)$$

where

$$\begin{cases} r0 = k_p + k_i + K_d \\ r1 = -(K_p + 2*K_d) \\ r2 = K_d \\ Ts = 0.05s \end{cases}$$

Programming the control law:
The steps of the algorithm are:
- Initialisation $u_0 = 0$, $\varepsilon_1 = 0$ and $\varepsilon_2 = 0$
- Read the measurement y
- Calculate the error $\varepsilon = y_c - y$
- Calculate and send the control law $u(k) = u_0 + r0.\varepsilon + r1.\varepsilon_1 + r2.\varepsilon_2$
- Save $u_0 = u(k)$, $\varepsilon_1 = \varepsilon(k)$ and $\varepsilon_2 = \varepsilon(k-1)$
- Wait until the end of the sampling period
- Go to step b.

5.2 Implementation of the neural network (multilayer perceptron) MLP

The steps of the algorithm of MLP are:
- Choice of the Software (Python)

9

- Choice of the model architecture (2 hidden layer NN)
- Arrays of weight vectors (random)
- Choice of activation function (sigmoid function)
- Choice of Learning method (supervised learning)
- Collection of a training data (input, desired output)
- Update weight vectors

In real time test, the following track was selected as the robot's evolution medium in order to validate the simulation as shown in Fig. 15.
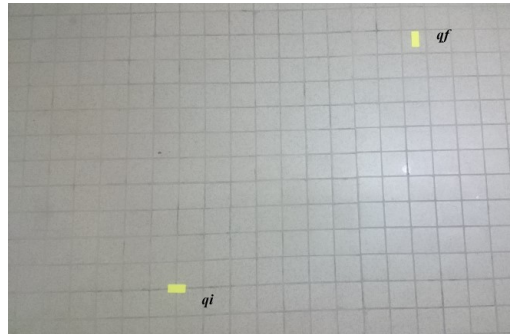

Figure 15: Evolution medium

With qi is the starting point and qf is the goal.

The evolution step of the robot in real time test are given in the figures bellow

$Step\,0$ is the initial position of the robot

$Step\,1$ present $\Delta y1$

$Step\,2$ present $\Delta y2$

$Step\,3$ present $\Delta y3$

5.3 Results and discussion

Step0
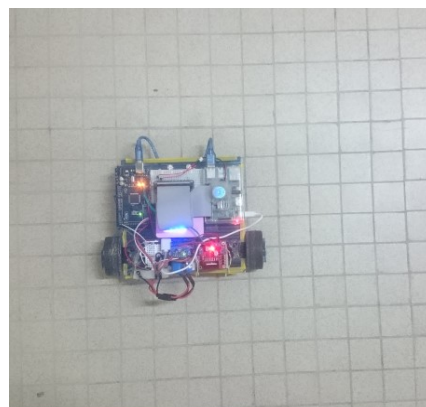
Figure 16: Robot position   Figure 17: Angular position
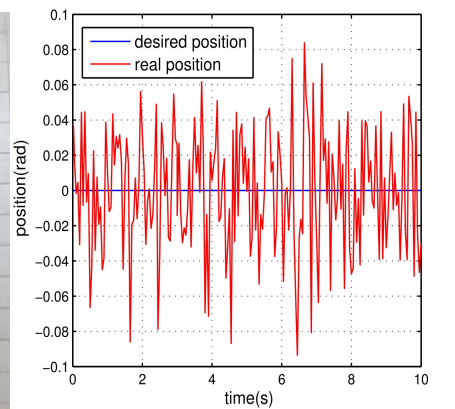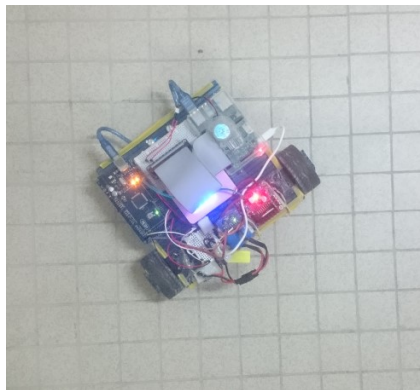
Step1

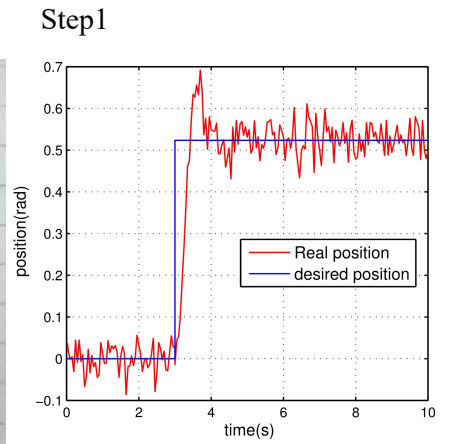Figure 18: Robot position

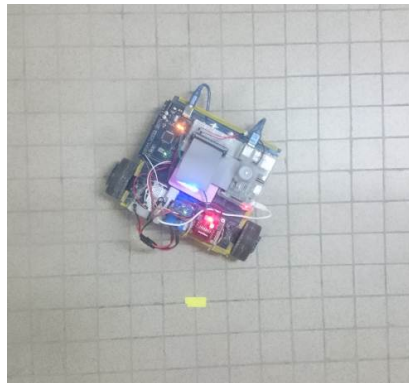Figure 19: Angular position

Step2
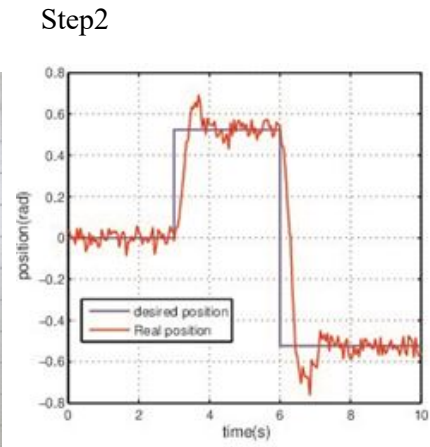

Figure 20: Robot position
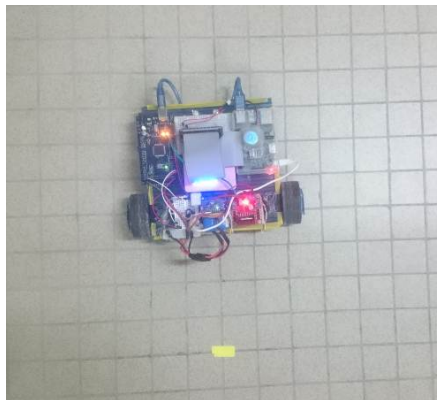

Figure 21: Angular position

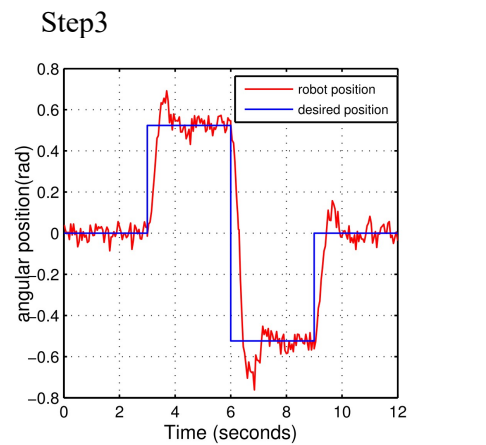## Step3

Figure 22: Robot position

Figure 23: Angular position

Now, the robot will redo these steps to complete its evolution on the y axis.

After moving, the robot has to turn $-\dfrac{\pi}{2}$ to reach his goal by moving Dx.
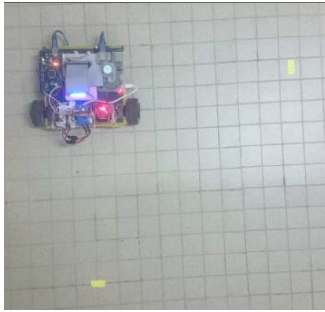
Step4
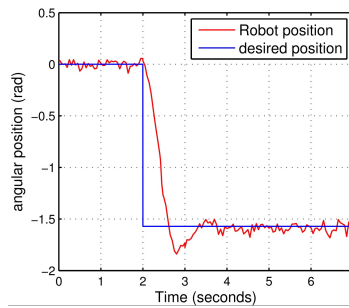

Figure 24: Robot position
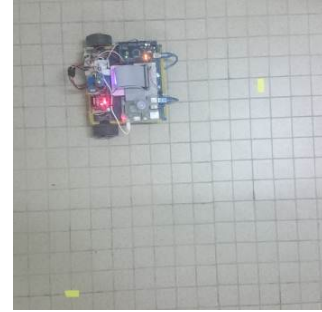

Figure 25: Angular position

Figure 26: Angular position

Now, the robot will start its path on the x-axis to achieve its goal and complete its mission.
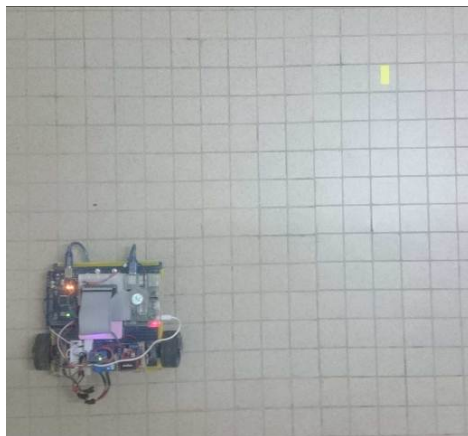

Figure 27: Initial position
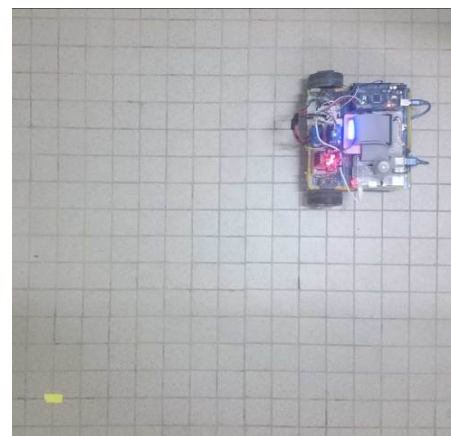
Figure 28: Final position

The strategy characterization are given in the table bellow

Characteristics	advantages or disadvantages
precision ++	++
complexity ++	++
Speed –	--
Power consumption ++	++

6. CONCLUSIONS

n this study, a new navigation strategy was developed and a real prototype was built to validate theory and simulation. The real-time tests showed good performance and high accuracy. As a

perspective, an attention will be focused on the generalization of the strategy for other types of mobile robots.

ACKNOWLEDGEMENTS

REFERENCES

[1] Borenstein, J., Experimental results from internal odometry error correction with the OmniMate mobile robot journal=IEEE Transactions on Robotics and Automation, In volume=14, number=6, pages=963-969,1998, publisher=IEEE.

[2] C. T. Sauer, H. Brugger, E. P. Hofer, and B. Tibken. Odometry error correction by sensor fusion for autonomous mobile robot navigation. 3:1654–1658, 2001.

[3] A. Martinelli. The odometry error of a mobile robot with a synchronous drive system. IEEE Transactions on Robotics and Automation, 18(3):399–405, 2002.

[4] J. Palacn Roca, J. A. Salse, Ignasi Valgan, and Xavier Clua. Building a mobile robot for a floor-cleaning operation in domestic environments. IEEE Transactions on Instrumentation and Measurement, 2004, vol. 53,p. 1418-1424, 2004.

[5] Johann Borenstein. The clapper: A dual-drive mobile robot with internal correction of dead-reckoning errors. pages 3085–3090, 1994.

[6] OKAMOTO Kenji. Ultrasonic doppler speed sensor with an offset parabolic reflector. JSAM, 63(2):53–59, 2001.

[7] Lin Zhao, Washington Y Ochieng, Mohammed A Quddus, and Robert B Noland. An extended kalman filter algorithm for integrating gps and low cost dead reckoning system data for vehicle performance and emissions monitoring. The journal of Navigation, 56(02):257–275, 2003.

[8] C.Rocken, R. Anthes, M. Exner, D. Hunt Analysis and validation of gps/met data in the neutral atmosphere. Journal of Geophysical Research: Atmospheres, 102(D25):29849–29866, 1997.

[9] Toshihiro Aono, Kenjiro Fujii, Shintaro Hatsumoto, and Takayuki Kamiya. Mobile Indicators in GIS and GPS Positioning Accuracy in Cities In Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, volume 4, pages 3443–3448. IEEE, 1998.

[10] Artur Janowski, Aleksander Nowak, Marek Przyborski, and Jakub Szulwic. Mobile Indicators in GIS and GPS Positioning Accuracy in Cities In Rough Sets and Intelligent Systems Paradigms, RSEISP 2014

Authors

Radwen Bahri was born in Tunisia. He is currently an Ph.D student in the National Engineering School of Gabes (Tunisia) and he is a member of the Modelling, Analysis and Control of Systems (MACS) laboratory. His current research interests include automatic control, robotics, and sensor design.

.