Neural controller of autonomous driving mobile robot by an embedded camera

Hajer Omrane
Department of Electrical Engineering
METS Laboratory
National Engineering School of Sfax, Tunisia
Hajeromrane2011@yahoo.com

Mohamed Slim Masmoudi and Mohamed Masmoudi
Department of Electrical Engineering
METS Laboratory
National Engineering School of Sfax, Tunisia
med_slim_mas@yahoo.fr
Mohamed.masmoudi@enis.tn

***Abstract*—The purpose of this paper is to build an autonomous RC Car that uses Artificial Neural Network (ANN) for control. It describes the theory behind the neural network and autonomous vehicles, and how a prototype with a camera as its only input can be designed to test and evaluate the algorithm capabilities. The ANN is a good algorithm that could help recognize patterns in an image, it can with a training set, containing 2000 images, classify an image with 96% of accuracy rate. The main contribution of this paper consists in using a single camera for navigation, possibly for obstacle avoidance.**

Keywords—RC Car; Raspberry PI3; camera; neural network

I. INTRODUCTION

Vision provides a large amount of data from the environment and allows for intelligent interaction. In robotics, over the past two decades, technological innovations in the manufacture of cameras and the evolution of computers have made it possible to integrate complex systems of vision in embedded systems, whether on mobile robots for autonomous navigation or on vehicles for driving assistance. Artificial vision is of particular importance because it provides the machine with the necessary capacities to react with its environment. The visual control enables to determine the control law that will bring the robot from its initial position to the target position. The choice of the model of the camera is of great importance for the navigation, the quality of the image as well as the results of the treatment that depend on it. The image processing system must be a real-time system to produce results within a time tolerated by the navigation applications.

Several research studies fall within this theme. Farooqui *et al* proposed a model that introduces a real-time live broadcast for a surveillance system using Raspberry Pi3 and its camera module to reduce the number of accidents [1].

Amruta *et al* developed a navigation algorithm and obstacle detection using raspberry Pi and a HD camera [2]. Filliat *et al* presented, in their study, one of the best-known research studies related to visual navigation in a natural environment. They used a video camera to provide information necessary for the location and navigation of the robot in its environment [3].

In this paper, we propose an autonomous driving mobile robot consisting of various subsystems: an input unit (camera) used to collect input data on navigation paths; a processing unit that manages several tasks such as receiving data from our training on the Raspberry Pi board and the prediction of the neural network and a power unit circuit. The main advantage of the proposed driving mobile robot consists in using a single camera for navigation which could be used also for obstacle avoidance.

The rest of this paper is organized as follows. The next section outlines the mathematical model of the robot. Then section III presents the proposed image processing approach. Section IV describes the experimental setup. Section V introduces the experimental results. Finally, section VI presents the conclusion.

II. KINEMATIC MODEL

The designed robot has the same configuration as a normal four-wheel car:

- Fixed rear wheels equipped with a differential mechanism,
- Front wheels ensuring the robot direction [4].

The centers of the front and rear wheels are respectively (X_f, Y_f) and (X_r, Y_r). The length between the front and rear wheels is l. The speed and the steering angle of the front wheels are v and $\emptyset$ respectively. The angle between the car orientation and the x-axis is θ (Figure 1).

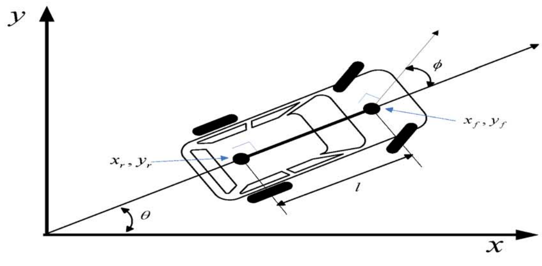


Fig.1. The kinematic model of the robot

From figure 1 the rolling condition of the non-slip rear wheels can be determined by [5]:

$$y_f \cos\theta - x_f \sin\theta = 0 \quad ,\tan\theta = \frac{y_f}{x_f} \quad (1)$$

Equation (1) identifies the tangential direction with any possible path for the vehicle and its movement along the curvature of the path.

According to the structure of the vehicle, one could find the centers of the rear wheels and the front wheels, such that, when one differentiate the following equations:

$$\begin{aligned} x_r &= x_f - l\cos\theta \\ y_r &= y_f - l\sin\theta \end{aligned} \quad (2)$$

One could obtain:

$$\begin{aligned} \dot{x}_r &= \dot{x}_f + l\dot{\theta}\sin\theta \\ \dot{y}_r &= \dot{y}_f - l\,\dot{\theta}\cos\theta \end{aligned} \quad (3)$$

By replacing Eq. (3) in Eq. (1), one could obtain:

$$\dot{x}_f \sin\theta - \dot{y}_f \cos\theta + l\dot{\theta} = 0 \quad (4)$$

According to Fig.1, the speed of the front wheels $(\dot{X}_f, \dot{Y}_f)$can be determined by:

$$\begin{aligned} \dot{x}_f &= v\cos(\theta + \phi) \\ \dot{y}_f &= v\sin(\theta + \phi) \end{aligned} \quad (5)$$

By replacing Eq. (5) in Eq. (4), one could obtain:

$$\dot{\theta} = \frac{v\sin(\phi)}{l} \quad (6)$$

Hence, the kinematic model of the front wheels would be then modeled by the above formulated Eqs (5) and (6).

The kinematic model of the rear wheels could be then derived from Eqs (5), (6) and (3) according to the following ones:

$$\begin{aligned} \dot{x}_r &= v\cos\theta\cos\phi \\ \dot{y}_r &= v\sin\theta\cos\phi \\ \dot{\theta} &= \frac{v\sin(\phi)}{l} \end{aligned} \quad (7)$$

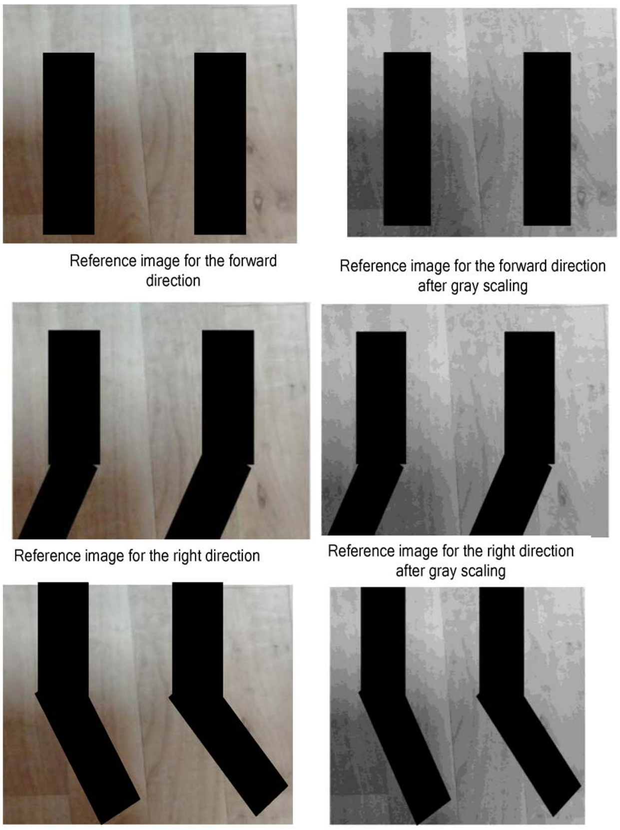
Using the kinematic models of the rear and front wheels, the next position of the robot can be calculated using the speed and angle of rotation of the front wheels $\emptyset$.

III. PROPOSED IMAGE PROCESSING AND NEURAL NETWORK CONTROLLER

A. Gray Scale image

The gray level of an image is simply designed to have colors that are all expressed in gray. Indeed, "gray" is a color in which all the components: red, green and blue have the same intensity in the RGB space (Red, Green and Blue)[6].

The following figures present some possible configuration of paths to be followed by the driving mobile robot during his navigation. They show the reference acquired image before and after transforming to gray level. The size of the acquired images is fixed to 320×120 pixels.



Reference image for the forward direction

Reference image for the forward direction after gray scaling

Reference image for the right direction

Reference image for the right direction after gray scaling

Reference image for the left direction

Reference image for the left direction after gray scaling

Fig.2. Gray scale processed image

B. The activation function

There are several machine learning algorithms, among which we can cite the Artificial Neural Networks (ANN) which are a fairly attractive technique for computer vision problems, particularly those of recognition. There is a big number neural network models which may vary according to the number of layers, neurons, learning algorithms, and so on.

To be sure that the ANN are reliable, the activation function is of great importance. Since it gives the sigmoid function, which is also called the logistic function [7] modeled according to the following equation:

$$g(x) = \frac{1}{1 + e^{-x}} \quad (8)$$

For the activation function, the sigmoid function have been chosen as a nonlinear function, since it presents real intermediate values between 0.05 and 0.95 which represents a good choice for solving nonlinear systems [8]. The images collected by the camera are converted into gray scale. We used only the lower half of the image size (320×120). There are 38,400 nodes in the input layer and 20 nodes in the hidden layer. There are four nodes in the output layer where each node corresponds to the steering control instructions: left, right, forward and reverse respectively. Figure 3 shows the structure of the neural network.

One advantage of using neural network is that once the network is trained, it only needs to load trained parameters afterwards, thus prediction can be very fast.

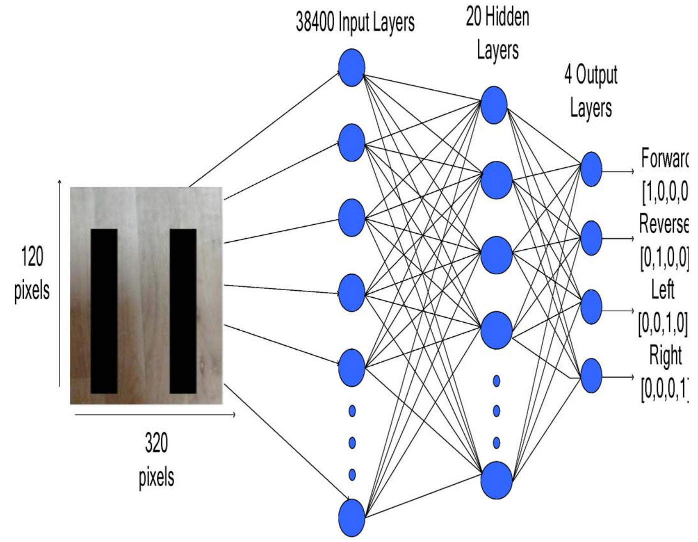


Fig.3. Structure of the neural network

To obtain a good generalization performance, one could use the logistic regression cost function which reflects the difference between the obtained output values and the reference values [9] as shown in equation (9).

The cost of a prediction with respect to the weights is a function J(W) modeled by:

$$J(W) = -\frac{1}{M}\left(\sum_{m=1}^{M} y_m \log g(x_m) + (1-y_m)\log\big(1-g(x_m)\big)\right) \quad (9)$$

where

M : Number of training examples used

J: Number of hidden-layer

w: Array of weights

x_m: Input to activation function

y_m: Output from the Neural Network

Since the required outputs from the training examples y_m is ϵ $\{0,1\}$, either of the terms is still zeroed out. This cost function was, and is often, selected for ANNs because it is convex which makes it more easily reducible than a non-convex cost function.

We optimized the cost function $J(W)$ by multiplying the gradient by the learning rate. The rate is chosen carefully to balance the speed of learning against that of overshooting the global minimum of the cost function [10].

The error term for a neuron of the output layer is calculated as follows:

$$\delta_i^{(3)} = a_i^{(3)} - y_i \quad (10)$$

The differentiation of $J(W)$ with respect to $W_{i,j}^{(2)}$ is then calculated as:

$$\frac{\partial J(W_{j,i}^{(2)})}{\partial (W_{j,i}^{(2)})} = \delta_i^{(3)} a_i^{(2)} \quad (11)$$

Hence, the error term $\delta_i^{(3)}$ is back-propagated to the hidden layer using Eq. (12). The error term of which is based on a weighted average of all the error terms in the output layer. This term is calculated for each neuron as:

$$\delta_j^{(2)} = W_{j,i}^{(2)} \delta_i^{(3)} h'(z_j^{(2)}) \quad (12)$$

IV. Experimental Setup

A. Bloc diagram of the system

The proposed hardware structure of the driving mobile robot consists of three subsystems: 1) an input unit (camera) which is used to collect data, 2) a processing unit based on a Raspberry Pi card, in which a necessary neural control algorithm the classification and prediction of the output control and 3) a motor unit consisting of two DC motors and a power interface based on the L298n circuit. Figure 4 presents the block diagram of the proposed system.

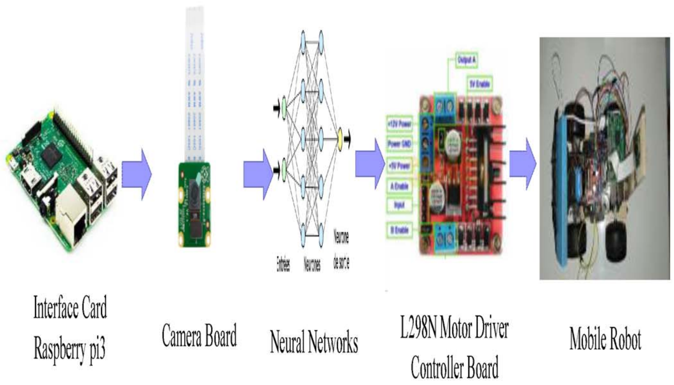


Fig.4. Block diagram of the system

B. The Raspberry PI3

The Raspberry Pi version B card consists of the Broadcom BCM2837 circuit that contains a 1.2GHz quad-core ARM Cortex-A5 processor [11]. Different peripherals can be connected to the card via USB ports or general purpose input pin (GPIO) inputs: Wi Fi USB, keyboard, mouse, etc. The card has an HDMI output that connects it to a monitor. The card runs a Raspbian operating system (derived from Linux) installed on an SD memory card.

C. PI Camera module

The RaspiCam camera is an Omnivision sensor which can acquire resolution images up to 3280 x 2464 and also support 1080p30, 720p60 and 640x480p90 videos with a frame rate of 15 fps in the YUV color space [12]. The RaspiCam camera weighs even more than 3g, making it perfect for mobile or other applications where the size and weight are important. Besides, it can be connected to the card using the Camera Serial Interface (CSI) port.

D. Raspbian OS

There are several different operating systems for the Raspberry Pi among which, the Raspbian stands out to be the most popular Debian Linux based operating system for theRaspberry Pi [13].

E. Installation Instructions

Setting Raspberry Pi to work with Matlab and Simulink software requires a support package to be installed onto both a computer and the board [14]. Command functions and special toolboxes are installed in the connected computer and a modified Raspbian OS with Matlab and Simulink compatibility is installed in Raspberry Pi.

F. Motor interfacing Board

L298N is a high voltage, high current motor driver chip, with the highest working voltage of 46V, continuous operating current of 2A, and instantaneous peak current up to 3A [15]. The chip contains two "H bridges" which are high voltage and high current full-bridge drivers that can directly drive two DC motors.

G. Mobile Robot

We will connect two GPIO pins from the Raspberry Pi to both the input pins of the L298n's Motor 1 and the output pins of motor 1 from L298n to the rear motor. With this approach we can control not only the forward/reverse direction of the motor, but also the motor speed by adjusting the frequency and the duty cycle of the pulse width modulation (PWM) output on the Enable pin. Moreover, two GPIO of the Raspberry Pi are connected to both the input pins of Motor 2 and the output pins of Motor 2 of the L298n are connected to the front motor.

V. EXPERIMENTAL RESULTS

A. Learning phase

We have performed the learning phase using MATLAB simulation. We have obtained the curves concerning the evolution of the error as a function of the number of iterations and the desired output with the output computed by the network. In the present study, we used neural networks with 2000 iterations that give us a good value of quadratic error (MSE) as illustrated on Fig. 6.

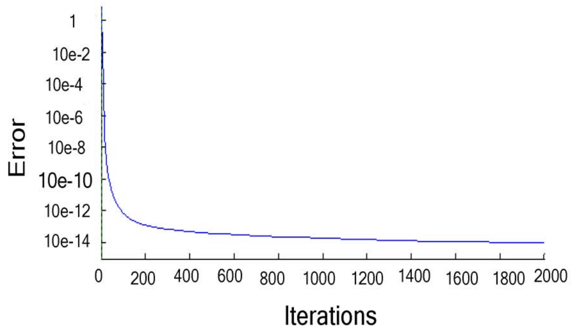


Fig.6. Learning curve

B. Generalization

Once the neural networks were trained (after learning), it is necessary to test it using databases different from those used during learning phase. This test helps both assess the performance of the neural system and detect the type of the problematic data. If the performance is not satisfactory, it will be necessary either to modify the architecture of the network, or modify the learning base. Figure 7 shows the generalization curve with 500 iterations which gives us a quadratic error (MSE) larger than the learning phase.

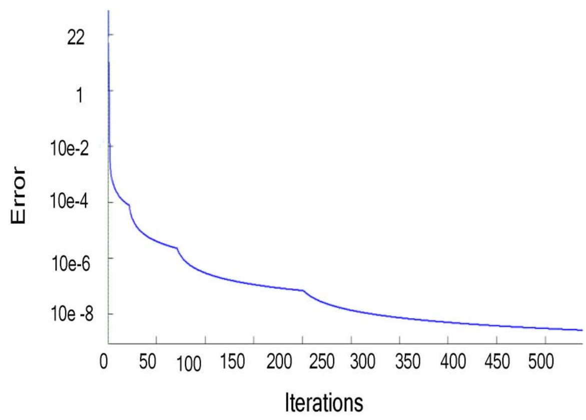


Fig.7. Generalization Curve

The simulation results show us that the larger the number of images provided in the learning stage, the more accurate the ANN will become. Table 1 shows the accuracy depending on the number of images in the training set.

TABLE I. Number of images in training set and Accuracy

Number of images	accuracy
180	74%
540	76%
900	78%
2000	96%

VI. CONCLUSION

The results obtained on a large number of successive images are generally satisfactory. The developed approach meets the settled requirements for autonomous navigation. Indeed the network of neurons is a good means of operation and control for an autonomous vehicle in order to have a high accuracy rate reaching 96%.The perspectives of our work are the improvement of the learning algorithm and the process of the experimental test.

References

[1] F. Shumaila Mateenuddin, Mrs. V.S. Jahagirdar, “An Android Controlled Mini Rover for real time surveillance using Raspberry Pi 3”, International Journal Of Advanced Research in Engineering & Management (IJAREM), 2017.

[2] A. Nikam1, A. Doddamani, D. Deshpande, S.Manjramkar, “RASPBERRY Pi BASED OBSTACLE AVOIDING ROBOT”, International Research Journal of Engineering and Technology, 2017.

[3] D. Filliat, “Navigation, perception and learning for robotics”, Proc. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 2014.

[4] D. Santos , G. Andouglas , S. Junior, “Design and Implementation of a Control System for a Sailboat Robot”,Robotic journal , 2016.

[5] W. Mark , S. Hutchinson, M. Vidyasagar, “Robot Modeling and Control”, Intelligent Systems & Agents,2016.

[6] https://en.wikipedia.org/wiki/Grayscale

[7] C. Cai, Y. Xu,“Deep Neural Networks with Multistate Activation Functions”, Computational Intelligence and Neuroscience, 2015.

[8] W. Bruce, E. Von Otter, “Artificial Neural Network Autonomous Vehicle”, Degree project technology, first cycle, 15 credits, stockholm sweden 2016.

[9] K.Hong, “ARTIFICIAL NEURAL NETWORK (ANN) 8 - DEEP LEARNING I : IMAGE RECOGNITION (IMAGE UPLOADING)”, Golden Gate Ave, San Francisco, USA, 2017

[10] S. raschka, “Machine Learning - Neural Networks Learning: Cost Function and Backpropagation”, Image Processing Mooc Machine Learning Algorithm, 2016

[11] https://www.mathworks.com/hardware-support/raspberry-pi-simulink.html

[12] G. Singh Pannu , M. Dawud , P. Gupta ,“Design and Implementation of Autonomous Car using Raspberry Pi”,International Journal of Computer Applications 2015.

[13] S. Amit , D. Sreedhar, B. Satyanaryana, “ Implementation of Linux Web Server for WebBased Automation System Using Raspberry Pi”, International Journal of Advanced Research in Computer and Communication Engineering, 2016.

[14] M. Mahesh , R. Yuvraj, “Obstacle Detection And Object Size Measurement For Autonomous Mobile Robot Using Sensor”, International Research Journal of Engineering and Technology ,2016.

[15] https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf