Trends in Applied Sciences Research

ISSN 1819-3579

Academic
Journals Inc.

www.academicjournals.com

Trends in Applied Sciences Research 9 (5): 246-253, 2014
ISSN 1819-3579 / DOI: 10.3923/tasr.2014.246.253

Neural Network Navigation Technique for Unmanned Vehicle

Boumediene Selma and Samira Chourqaui
Department of Computer Science, Faculty of Science, University of Science and Technology, "Mohamed Boudiaf" USTO Oran, 31000, Algeria

Corresponding Author: Boumediene Selma, Department of Computer Science, Faculty of Science, University of Science and Technology, "Mohamed Boudiaf" USTO Oran, 31000, Algeria Tel: +213776653511

ABSTRACT

This study deals with a path planning and intelligent control of an autonomous vehicle which should move safely in its road partially structured. This road involves a number of obstacles like speed bump, traffic lights and other vehicles. In this study the Neural Networks-based technique (NN) is described to solve the motion-planning problem in Unmanned Vehicle (UV) control. This is accomplished by choosing the appropriate inputs/outputs and by carefully training the NN. The network is supplied with distances of the closest obstacles around the vehicle to imitate what a human driver would see. The output is the acceleration and steering of the vehicle. The network has been trained with a set of strategic input-output. The results show the effectiveness of the technique used, the Unmanned Vehicle (UV) drives around avoiding obstacles.

Key words: Unmanned vehicle, neural network, control

INTRODUCTION

Unmanned Vehicle (UV) has the ability of a mobile robot to reach the set targets by avoiding obstacles in its way. Thus essential behaviors for UV navigation are obstacle avoidance and goal reaching (Sam Ge and Lewis, 2006; Cuesta and Ollero, 2005). Conventional control techniques can be used to build controllers for these behaviors; however, the environment uncertainty imposes a serious problem in developing the complete mathematical model of the system resulting in limited usability of these controllers. Thus some kind of intelligent controllers are required that can cope with the changing environment conditions. Amongst the various artificial intelligence techniques available in literature, neural networks offer promising solution to autonomous navigation problem because of their ability to learn complex non linear relationships between input values and output control variables. This ability of neural networks has attracted many researchers across the globe in developing neural network based controllers for reactive navigation of mobile robots or vehicles in its environments.

The proposed study is to have a vehicle that drives by itself and avoids obstacles in a virtual world. Every instant, the vehicle decides by itself how to modify its speed and direction according to its environment. In order to make it more real, the NN should only see what a person would see if it was driving, so the UV decision is only based on obstacles that are in front of the vehicle. By having a realistic input, the NN could possibly be used in a real vehicle and work just as well.

Many driving games could use this technique to control vehicles, but there are a number of other applications that could be found for software that controls a vehicle in a virtual world, or in the real world, some of the techniques proposed include fuzzy control (Banks and Hayward, 2001; Doitsidis *et al*., 2004; Verbruggen *et al*., 1999), adaptive control (Andrievsky and Fradkov, 2002; Astrom and Wittenmark, 1989; Schumacher and Kumar, 2000), neural networks (Li *et al*., 2001), genetic algorithms (Cordon *et al*., 2004), lyapunov theory (Ren and Beard, 2003) and neuro-fuzzy control (Selma and Chouraqui, 2012). Neural networks are based on how our brain works; they seem to be the right choice.

MATERIALS AND METHODS

The objective of the navigation system developed in this work consists of driving the vehicle to follow a reference path from an initial point to a final one in a partially structured environment as shown in Fig. 1. Unexpected fixed obstacles are considered, showed in Fig. 2.

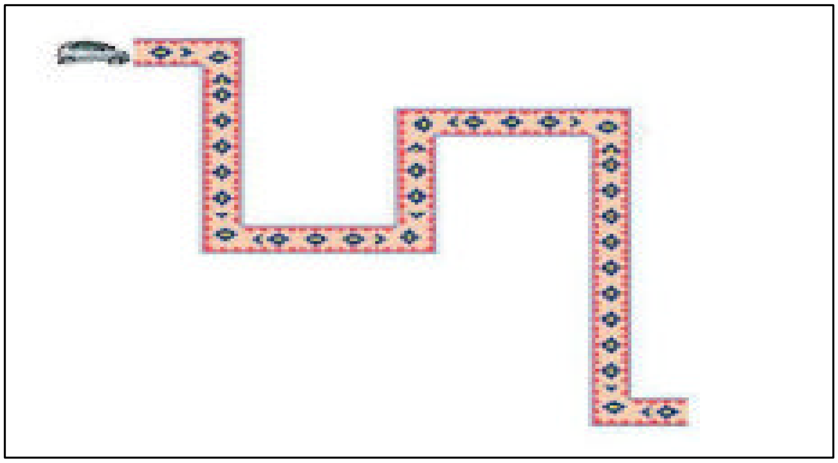


Fig. 1: Reference path

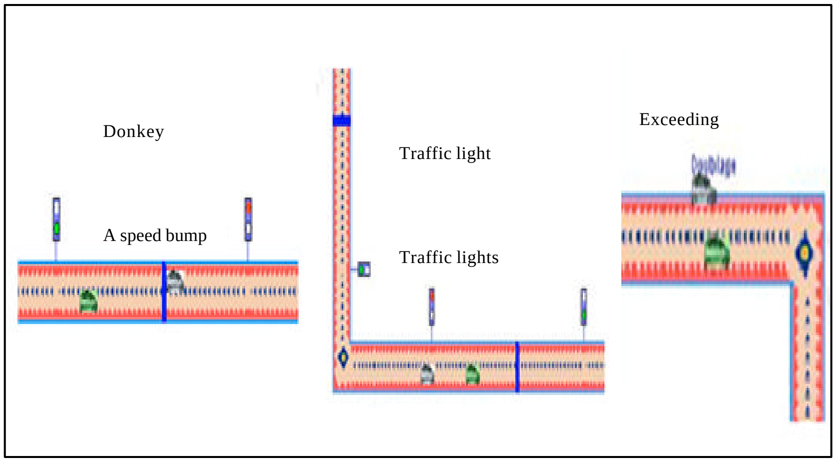


Fig. 2: Obstacles avoidance (Speed bump, traffic lights and vehicles exceeding)

NEURAL NETWORKS

The neural network artificial intelligence comes from how brains work. This principle is applied to neural networks on a smaller scale. Today's computers don't have the power of computing that 20 billion neurons do, but even with a few neurons, we are able to have intelligent response from a neural network.

Neurons are organized in layers as Fig. 1 shows. The input layer will have entries and depending on the strength of connection to each neuron in the next layer, the input signal is sent to the next layer. The strength of the connection is called a weight. The value of each neuron in each layer will depend on the weight of the connection and the values of the neurons of the previous layer.

A driver could be compared to a "function". There are different inputs: What the driver sees. This data is processed by the brain as a function and the response of the driver is the output of the function.

A function f(x) = y transforms a value x (one dimension) into y (one dimension).

A back propagation neural network is used for the "brain" of the driver because such a neural network is capable of approximating any function with a domain and a range that each has multiple dimensions: $f(x_1, x_2,..., x_n) = y_1, y_2,..., y_m$.

That is exactly what is needed here since multiple inputs and multiple outputs are implemented.

When a neural network has just a couple of neurons, it is simple to compute what the weight should be to get the desired response. But as the number of neurons is increased, the complexity of computing what the weight should be is augmented. A back propagation network lets us train the neural network which sets the weights for us. It is just needed to provide the desired outputs with the corresponding inputs.

After being trained, the neural network will respond close to the desired output when a known input is given and will "guess" a response for any input that doesn't exactly match the trained input-output.

The neural network used in this case has 3 layers (Fig. 3). The input layer has 4 neurons, and the output layer has 3. I'll explain why later. The layer in between have 5 neurons each.

When choosing the number of neurons, keep in mind that each layer and each neuron you add to the system will add computation time to calculate the weights.

Input: What information is important in controlling a vehicle in its road? Firstly, it is important to know the position of road and position of obstacles and velocity? If there are buildings on both side of the road, but none if front, the vehicle will accelerate. But if a car is stopped in front, it will break. Secondly, the distance from vehicle position to the object. If an object is far away it will continue driving until it is close, in which case, it will slowdown or stop.

That is exactly the information that are used for the proposed neural network. To keep it simple, four positions x,y, obstacles and the velocity of the vehicle are considered.

To be able to do this, the position (x,y) of every object, the position (x,y) of the road, the position (x,y) of the vehicle and the position (x,y) of obstacles are measured.

Because the neural network is using a sigmoid function which is defined by Eq.1:

$$v_j = \frac{e^{z_j}}{e^{z_j}+1} = \frac{1}{1+e^{-z_j}},\ \ j= 1,...,m \tag{1}$$

The input needs to be between 0.0 and 1.0.

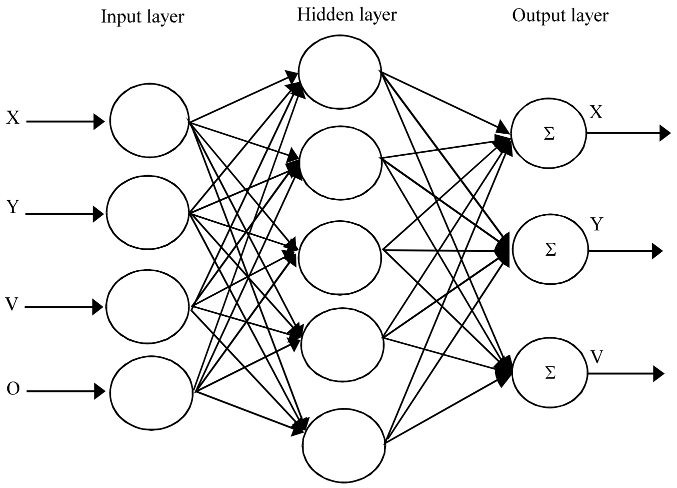
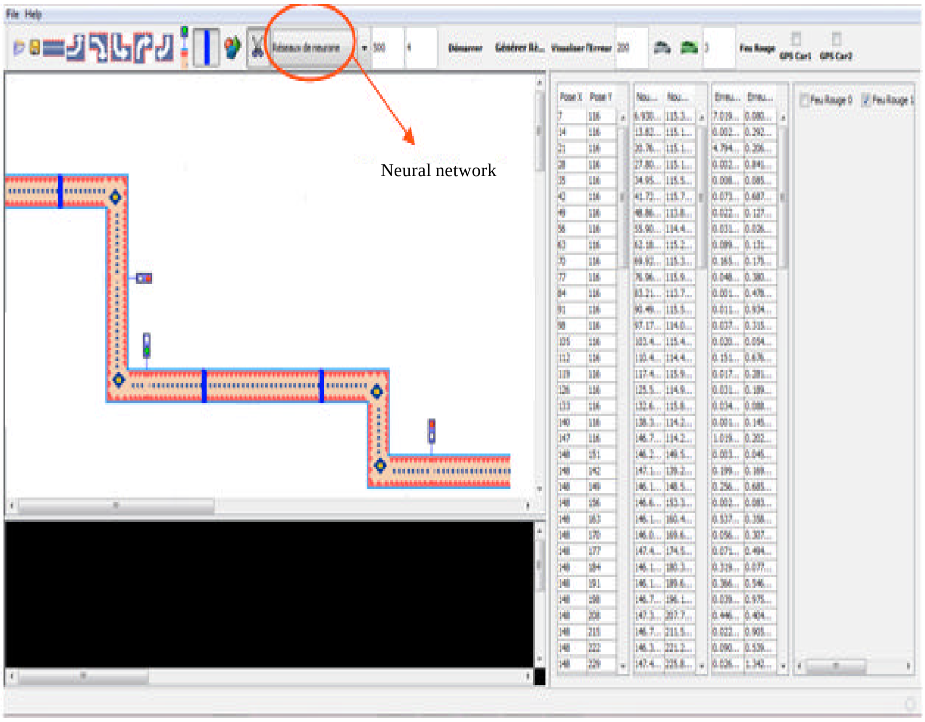


Fig. 3: Neural network architecture



Pose X	Pose Y	Nou...	Nou...	Erreu...	Erreu...
7	116	6.930...	115.3...	7.019...	0.080...
14	116	13.82...	115.1...	0.002...	0.292...
21	116	20.76...	115.1...	4.794...	0.206...
28	116	27.80...	115.1...	0.002...	0.841...
35	116	34.95...	115.5...	0.008...	0.085...
42	116	41.72...	115.7...	0.073...	0.687...
49	116	48.86...	113.8...	0.022...	0.127...
56	116	55.90...	114.4...	0.031...	0.026...
63	116	62.18...	115.2...	0.089...	0.131...
70	116	69.92...	115.3...	0.165...	0.175...
77	116	76.96...	115.9...	0.048...	0.380...
84	116	83.21...	113.7...	0.001...	0.478...
91	116	90.49...	115.5...	0.011...	0.934...
98	116	97.17...	114.0...	0.037...	0.315...
105	116	103.4...	115.4...	0.020...	0.054...
112	116	110.4...	114.4...	0.151...	0.676...
119	116	117.4...	115.9...	0.017...	0.281...
126	116	125.5...	114.9...	0.031...	0.189...
133	116	132.6...	115.8...	0.034...	0.088...
140	116	138.3...	114.2...	0.001...	0.145...
147	116	146.7...	114.2...	1.019...	0.202...
148	151	146.2...	146.5...	0.003...	0.045...
148	142	147.1...	139.2...	0.199...	0.169...
148	149	146.1...	148.5...	0.256...	0.685...
148	156	146.6...	153.3...	0.002...	0.083...
148	163	146.1...	160.4...	0.537...	0.358...
148	170	146.0...	169.6...	0.056...	0.307...
148	177	147.4...	174.5...	0.071...	0.494...
148	184	146.1...	180.3...	0.319...	0.077...
148	191	146.1...	189.6...	0.366...	0.546...
148	198	146.7...	196.1...	0.039...	0.975...
148	208	147.3...	207.7...	0.446...	0.404...
148	215	146.7...	211.5...	0.022...	0.905...
148	222	146.3...	221.2...	0.090...	0.539...
148	229	147.4...	225.8...	0.026...	1.342...

Fig. 4: Learning about this sample reference path

Output: The output needs to control the vehicle's speed and direction. That would be the acceleration, the brake and the steering wheel. So 3 outputs are needed; one will be the acceleration/brake since the brake is just a negative acceleration, and the others will be the position. The output from the neural network is also between 0.0 and 1.0 as the input.

As it is illustrated in Figure 3, the initial positions of the reference route (X,Y), the velocity V, back to Speed bump D, F traffic light, vehicle O which shows the state of overshoot are in the entry of the network and three outputs, new positions (X,Y) and the velocity V.

Table 1: UV situations on the road

Input neurons relative to obstacle and velocity			Output neurons relative to velocity	
Obstacle	Velocity	Antecedent	Acceleration	Consequent
No obstacle	Full acceleration	/	Full acceleration	No action
No obstacle	Low acceleration	/	Accelerate	/
Donkey	Full acceleration	Distance speed bump is D	Slow down	/
Traffic lights	Full acceleration	Color traffic lights is green	Slow down	/
Traffic lights	Full acceleration	Color traffic lights is red	Stop	/
Vehicle	Full acceleration	Our vehicle speed is greater than the vehicle on the road	Full acceleration	Change of position and exceeding
Vehicle	Full acceleration	Speed of your vehicle is smaller than that of vehicle in road	Full acceleration	No action

According to the calculations of the network (back propagation algorithm) the control action is applied to the input of the process that lets you know the state of the system to calculate the error which has the input of the network controller.

Training: As it is mentioned earlier, training the neural network is required to approximate the function to be accomplished. A set of input-output is generated that would be the main reaction that the neural network have to follow.

Choosing the right input-output to train a neural network is probably the trickiest part of all. I had to train the network with a set of input-output, see how it reacted in an environment, and modify the entries as needed. Depending on how the network is trained, the vehicle can hesitate in some situations and get immobilized.

In Table 1 different arrangements of obstacle relative to the vehicle, velocity and the desired reaction from the UV are visualized.

SIMULATION RESULT AND DISCUSSION

To show the contribution of the control by ANN, simulation was approved on an unmanned vehicle.

Figure 4 shows the path followed by the UV controlled by the proposed ANN including obstacles described in the above sections.

Figure 5 shows the vehicle's path obtained after controlling with the neural network and Fig. 6 give the error calculated between the reference path obtained one after application of control.

As it can be seen from Fig. 5 and 6 the path obtained from simulation setup is more close to the reference path which validates the proposed method.

Figure 7 shows the generations of rules. The function of the speed controller subsystem control is to achieve the desired speed that is to say the increase and decrease in speed on the road and especially in front of obstacles, so the accelerator control. Figure 8 shows the variation of the UV speed in function of obstacles come across the road.

From the Fig. 8, it can be seen that the UV can indeed avoid obstacles and reach the targets. To verify the feasibility of proposed method Table 2 shows results of neural network controller.

Therefore, it can be concluded that the NN controller have a good potential to effect fast response to obstacles and reduce errors (Table 3).

Table 2: NN best simulation results

	Statistical results	
Datasets	Rate accuracy	Total classification accuracy (%)
Set X	98.43	98.77
Set Y	99.11	

Table 3: Comparison of the NN results with GA-FL

Approach	Average error
NN	1.23
GA-FL	3.81

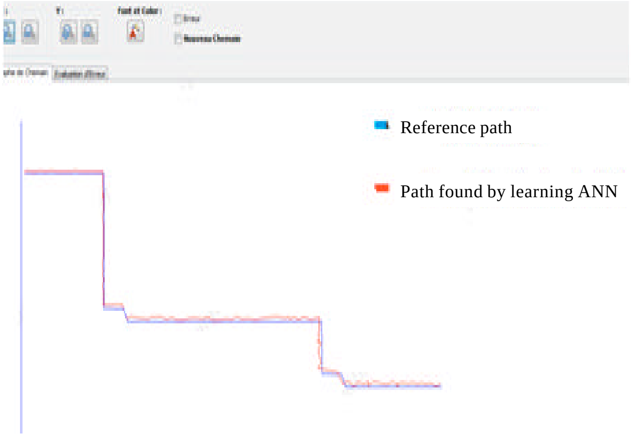


Fig. 5: Path found by the neural networks

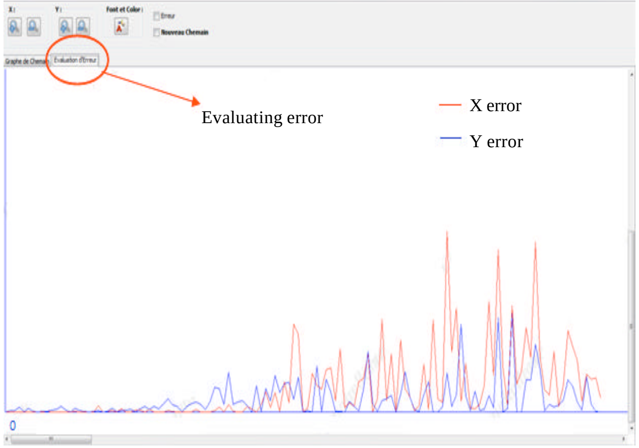
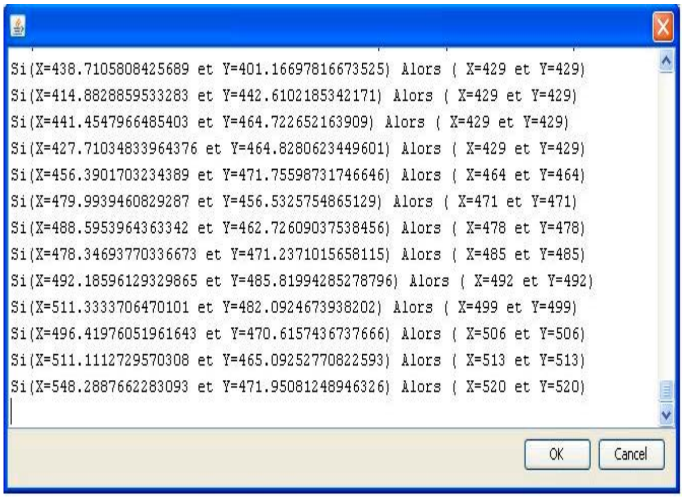


Fig. 6: Error between the reference path and the path calculated



```
Si(X=438.7105808425689 et Y=401.16697816673525) Alors ( X=429 et Y=429)
Si(X=414.8828859533283 et Y=442.6102185342171) Alors ( X=429 et Y=429)
Si(X=441.4547966485403 et Y=464.722652163909) Alors ( X=429 et Y=429)
Si(X=427.71034833964376 et Y=464.8280623449601) Alors ( X=429 et Y=429)
Si(X=456.3901703234389 et Y=471.7559873174664 6) Alors ( X=464 et Y=464)
Si(X=479.9939460829287 et Y=456.5325754865129) Alors ( X=471 et Y=471)
Si(X=488.5953964363342 et Y=462.72609037538456) Alors ( X=478 et Y=478)
Si(X=478.34693770336673 et Y=471.2371015658115) Alors ( X=485 et Y=485)
Si(X=492.18596129329865 et Y=485.81994285278796) Alors ( X=492 et Y=492)
Si(X=511.3333706470101 et Y=482.0924673938202) Alors ( X=499 et Y=499)
Si(X=496.41976051961643 et Y=470.6157436737666) Alors ( X=506 et Y=506)
Si(X=511.1112729570308 et Y=465.09252770822593) Alors ( X=513 et Y=513)
Si(X=548.2887662283093 et Y=471.95081248946326) Alors ( X=520 et Y=520)
```

Fig. 7: Generation of rules

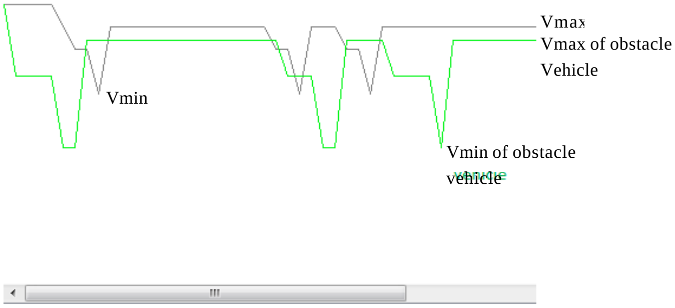


Fig. 8: Vehicle speed variations at obstacles

To show the performance of the results obtained by our approach, an approach based on type-2 fuzzy logic theory and genetic algorithms (Martinez *et al*., 2009) have been selected for comparison because of its high capacity of prediction and control in non linear dynamical systems.

CONCLUSION

We have designed a trajectory tracking controller taking into account the obstacles using neural network. NN architecture has demonstrated a good performance in modeling the trajectory of an Unmanned Vehicle.

Until now we have good results in the control, because one of the objectives was the vehicle stability on a desired trajectory.

Through experimental tests, it is found that the proposed system is a powerful tool for controlling non linear dynamical systems.

REFERENCES

Andrievsky, B. and A. Fradkov, 2002. Combined adaptive autopilot for an UAV flight control. Proceedings of the International Conference on Control Applications, Volume 1, September 18-20, 2002, Glasgow, Scotland, pp: 290-291.

Astrom, K.J. and B. Wittenmark, 1989. Adaptive Control. Addison Wesley Publishing Company, Reading, USA.

Banks, W. and G. Hayward, 2001. Fuzzy logic in embedded microcomputers and control systems. Byte Craft Limited, Ontario, Canada. http://www.bytecraft.com/fuzlogic.pdf.

Cordon, O., F. Gomide, F. Herrera, F. Hoffmann and L. Magdalena, 2004. Ten years of genetic fuzzy systems: Current framework and new trends. Fuzzy Sets Syst., 141: 5-31.

Cuesta, F. and A. Ollero, 2005. Intelligent Mobile Robot Navigation. Springer, The Netherlands, ISBN: 978-3-540-23956-7, Pages: 204.

Doitsidis, L., K.P. Valavanis, N.C. Tsourveloudis and M. Kontitsis, 2004. A framework for fuzzy logic based UAV navigation and control. Proceedings of the IEEE International Conference on Robotics and Automation, Volume 4, April 26-May 1, 2004, New Orleans, LA., USA., pp: 4041-4046.

Li, Y., N. Sundararajan and P. Saratchandran, 2001. Neuro-controller design for nonlinear fighter aircraft maneuver using fully tuned RBF networks. Automatica, 37: 1293-1301.

Martinez, R., O. Castillo and L.T. Aguilar, 2009. Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. Inform. Sci., 179: 2158-2174.

Ren, W. and R.W. Beard, 2003. CLF-based tracking control for UAV kinematic models with saturation constraints. Proceedings of the 42nd IEEE Conference on Decision and Control, Volume 4, December 9-21, 2003, Provo, UT, USA., pp: 3924-3929.

Sam Ge, S. and F.L. Lewis, 2006. Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications. CRC Press, Boca Raton, ISBN-13: 9781420019445, Pages: 736.

Schumacher, C.J. and R. Kumar, 2000. Adaptive control of UAVs in close-coupled formation flight. Proceedings of the American Control Conference, Volume 2, June 28-30, 2000, Chicago, IL., pp: 849-853.

Selma, B. and S. Chouraqui, 2012. Trajectory estimation and control of vehicle using neuro-fuzzy technique. Int. J. Adv. Eng. Technol., 3: 97-107.

Verbruggen, H.B., H.J. Zimmerman and R. Babuska, 1999. Fuzzy Algorithms for Control. 2nd Edn., Kluwer Academic Publishers, USA., ISBN: 9780792384618, Pages: 352.