

Neuro-Wireless Vehicle's Movement Control

Dr.Asmaa Q. Shareef

Baghdad University\College of Science
Computer Science Department

Abstract:

The aim of this paper is to control the movements of a vehicle wireless by using two ways: either to move in the four main directions through the keyboard of a computer, or to force the vehicle to move through a desired path by training the multi-layer feed forward neural network. This was done under three phases; the first phase was to design a specific hardware represented by an electronic board, which represents the transmission circuit to be interfaced with the computer to control the movement of the vehicle wirelessly, while the second phase was to implement a specific software to control the movement which was programmed by using the visual basic object oriented language, the third phase was to train the multi-layer feed forward neural network by the backpropagation learning algorithm to estimate the desired path of the vehicle's movement, then to control the movement of the vehicle according to the desired path.

Key words: Neuro-wierless, feed forward neural network, backpropagation, Control, Vehicle, visual basic ver.6, hardware, software

1-Introduction:

The wireless control system takes a special importance among other systems that would avoid human the danger of visiting danger places, like caves for example. The principle work of the wireless controlled vehicle depends on transmitting known signals from the remote control to the receiving circuit fixed in the vehicle, these signals will be received by the vehicle through the antenna, and each signal will trigger a pre-specified movement. The remote control contains two switches: one gives the directions forward and backward, the second gives the directions left and right. These switches are connected to other switches that activate the DC motors in a vehicle according to the required movement sequence.

Also the last century had witnessed a very enormous progress in all sciences, especially computers. The programming languages were one of the main fields in software (S/W) benefits from the great changing and progress that occurred. One of the most popular languages that the programmers used a lot was, GwBasic. This language had witnessed a lot of changing through next versions, QBasic, which is considered as a mother language or core for the next version of basic which was, visual basic (VB). VB presented a lot of facilities for the

programmers, especially it is working under windows, and also it is Object Oriented Language (OOL) that deals with objects. It has a very wide capabilities, one of them is interfacing. The flexibility of VB made it as one of the most languages that is used for interface.

Works related to the idea of the work in this paper were proposed in several cases as [1] described the concept of the navigation system for a mobile robot which used a self-learning feed forward neural network to generate initial path necessary to form a movement plan for a robot, then algorithm is placed into a PC, which is connected to mobile robot by wireless and wired links. [2] proposed a decentralized control method for controlling a platoon of N vehicles and a centralized controller that has real information about every vehicle. [3] This paper discussed a new idea for autonomous transportation system-intelligent smart cars. These cars can take some of the decision of their own. [4] stated the basic principles of the mathematical 3D modeling and simulation of the Tesla's wireless controlled moving vehicle Tesla boat. [5] designed a controller based on a mathematical model to control the quad-copter taking off and landing automatically, and a wireless communication system was implemented to enable the quad-copter receiving instructions from the controller and carry them out. [6] presented connection of real cars model, made in scale(1:4), main energy source was battery and direct methanol fuel cell. The vehicle remotely controlled by wireless radio frequency system (ZigBee). [7] proposed a control strategy of Automated Guided Vehicle (AGV), the vehicle - has three wheels- movement was controlled by an inboard PLC that do not need physical guide. The strategy is based on 2 main purposes: the path was stored in the PLC memory and the vehicle displacement was calculated from the wheel rotation measurement. The comparison between the required path and the actual position of the AGV allow calculating deviation error. Function of this error, a correction strategy of driving speed and steering angle was applied in order to get a smooth and precise displacement. [8] presented the design and implementation of a multiple robot motion control system for a wireless networking research using medium scale robots as couriers to the implementation of the system in a real world environment. [9] developed, implemented and tested planar motion control algorithms and adapted the robot for wireless control sensing techniques, an omnidirectional mobile is a special class of mobile robots platform that was built at the National University of Singapore (NUS). [10] mounted a camera on the vehicle (Quadroto) with the target in the field of view. An algorithm (least-squares estimations) processed the images, extracted the attitude and position information of the camera relative to the target utilizing geometry and four single-point discrete Fourier transforms on the moiré patterns. [11] designed and implemented a PIC controlled IR system to control the car's mirror system movement. The designed and built

system allows the side view mirrors to be adjusted based on the driver head movement. [12] presented the designed, development, and testing of a hardware (H/W) testbed for formation movement, using a fleet of autonomous remote control (R/C) vehicles equipped with Global Positioning Systems (GPS). Interface the remote of the R/C car to the centralized computer through the use of a DAC. [13] discussed intelligent-vehicle: Out-vehicle that involves collecting information about the driving environment sensing, and Vehicle-state sensing that focuses on measuring a vehicle's movement and monitoring its actuators. [14] introduced their developed mobile robot F.A.A.K., which utilizes several DSPs for its control and data processing, also presented a neuro-fuzzy controller for trajectory following suitable in space restricted applications like mobile robots.

In this paper design and implementation of a neuro-wireless movement control strategy for a vehicle is proposed. It is organized as follows. The system design is explained in section 2. Section 3 deals with the vehicle's parts. Section 4 deals with a description of H/W interface. In section 5, the Program interface is described; section 6 presents the components used in the designed interface circuit. In section 7 S/W implementation is presented, and the conclusion and suggestions for future work are given in section 8.

2-Proposed designed system

In this work, a vehicle (toy) was controlled to be moved by using the computer's keyboard either in straight lines or to be moved according to an estimated path produced as a response to the backpropagation (BP) learning algorithm that was applied to the feed forward neural network¹⁵ (FFNN) to obtain the desired paths. Fig.(1) represents the designed system that includes the designed circuit to be interfaced to the computer to control vehicle's movement. The designed system contained three major pieces: the H/W, the S/W, and the integration of the S/W and the H/W into a single system.

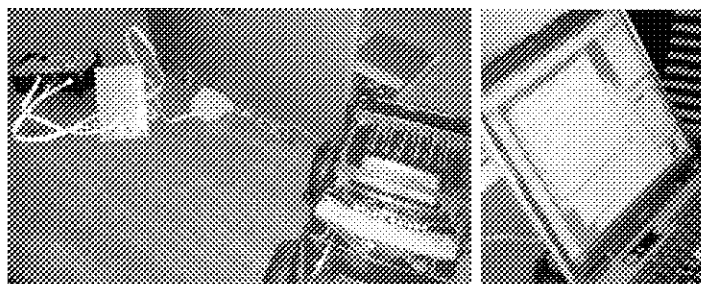


Fig.(1) The wireless designed system

The phase that controls the vehicle movement through the keyboard, simply reads the pressed arrow then sent response to the parallel port¹⁶ that interfaced to the designed electronic circuit. In the phase that uses the path that produced from the trained neural network (NN), this process occurs: each point was buffered in an array, and the distance between each two points was buffered

in another array, then it was programmed if the star button was pressed, in a serial movement will be sent to the parallel port, in response the vehicle will move according to a desired path. This phase is an open-loop control system that is designed to obtain the desired response without feedback, i.e. not to store the values of the trained NN. This is shown in Fig.(2).

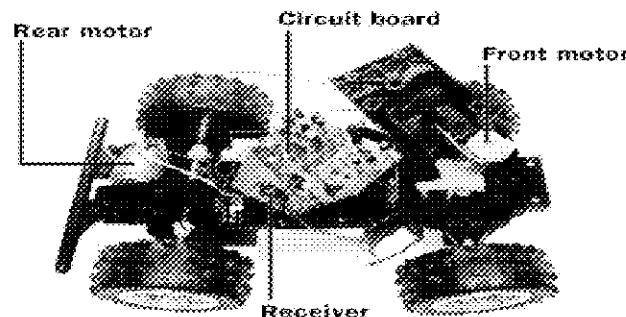


Fig.(2) The phase with NN

3-Vehicle's parts

The vehicle used in this project consists of two DC motors and one receiving transmission circuit. The first DC motor is fixed forward the vehicle. It is responsible of the left, right turning movements of the front tires. This motor is connected to gearbox that accelerates the response of the tires. The second DC motor is fixed at the backward the vehicle. It is responsible of the movement of the back tires forwarded and backward, it is also connected to a gearbox. The vehicle contains a 9 volt source which supplies the motors and the receiving circuit. Fig.(3) describes the above parts.

The transmission device (remote control) contains a 9 volt source, transmits a 27 MHz frequency. It contains two switches; one for driving vehicle forward or backward, and the second for turning left or right.



4-Hardware interface

A special circuit was designed to control the movement of the vehicle, each element of the circuit had been chosen carefully so that the circuit will work properly as desired. This circuit will initiate the transmitting from the remote control to the vehicle obeying signals from the computer that specifies the wanted position. The produced signal will pass through parallel port LPT1¹⁷ to the resistance then to the transistor that work as switch. In this paper, the LPT1 which is a parallel communication physical interface for connecting various peripherals is used as an output port for the signals to the interfaced circuit that will control the vehicle. The IEEE 1284 standard defines the bi-directional version of the port, which allows the transmission and reception of data bits at the same time. It also known as a printer port, it doesn't require

serial-to-parallel converter. It has 25 pin, as illustrated in Fig.(4). The pins 18 to 25 were connected to computer ground (0 volt).

4.1-Port addresses

Traditionally IBM PC systems have allocated their first three parallel ports according to the configuration in Table (1)¹⁷.

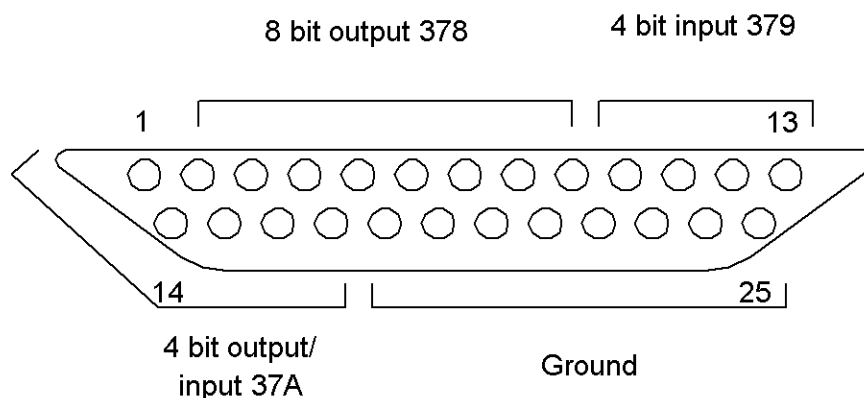


Fig.(4) The 25– pin female LPT1 parallel port¹⁷
Table (1)¹⁷

PORT NAME	Interrupt #	Starting I/O	Ending I/O
LPT1	IRQ 7	0x378	0x37F
LPT2	IRQ 5	0x278	0x27F
LPT3	IRQ 2	0x3bc	0x3bf

Table (2) for bit to pin mapping for Standard Parallel Port (SPP)¹⁶.

Address	MSB							LSB
	Bit: 7	6	5	4	3	2	1	0
Base (Data port)	Pin: 9	8	7	6	5	4	3	2

Table (2)¹⁶

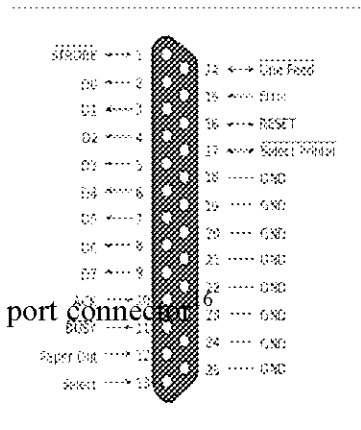
5-Program interface

In versions of Windows that did not use the Windows NT kernel (as well as DOS and some other operating systems), programs could access the parallel port with simple outportb() and inportb() subroutine commands, therefore in implementation the program is enforced to add Dynamic Library Link¹⁸ (DLL) library that prepares these procedures. In operating systems such as Windows NT and Unix, the microprocessor is operated in a different security ring, and access to the parallel port is inhibited, unless using the required driver¹⁶. Pin outs for parallel port connectors are listed in Table (3)¹⁶ related to Fig.(5).

Pin No (DB26)	Pin No (36 pin)	Signal name	Direction	Register bit
1	1	Strobe	In/Out	Control-0
2	2	Data0	Out	Data-0
3	3	Data1	Out	Data-1
4	4	Data2	Out	Data-2
5	5	Data3	Out	Data-3
6	6	Data4	Out	Data-4
7	7	Data5	Out	Data-5
8	8	Data6	Out	Data-6
9	9	Data7	Out	Data-7

Table (3)

Fig.(5) Pins out for parallel port connector



6-The Components used in the designed interface Circuit:

At the first steps of design, Fig.(6) was built, but a problem arised: that was in using a power supply to supply the designed circuite in the whole system as shown in Fig.(7-a), so in order not to use the power suply as shown in Fig.(7-b), the following componetes were substuted:

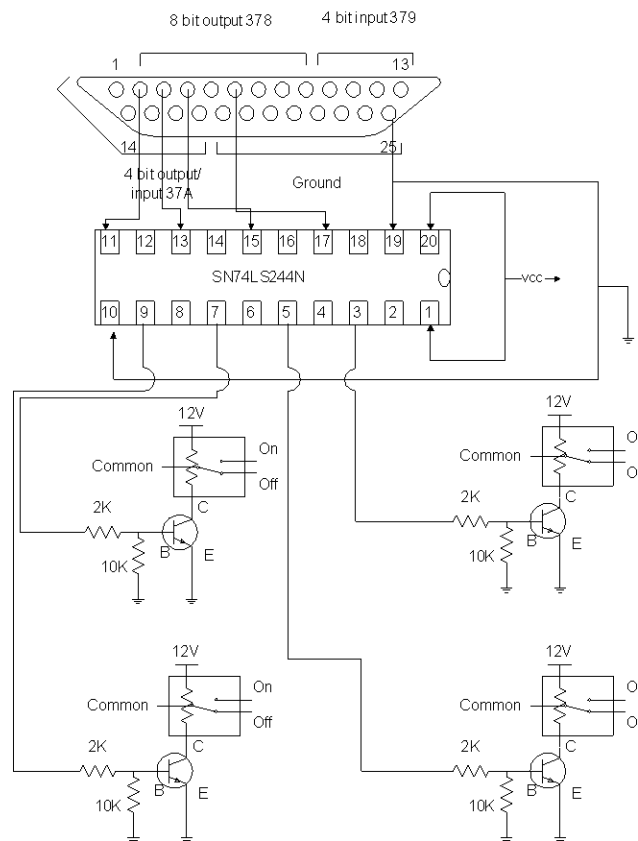


Fig.(6) The designed circuit

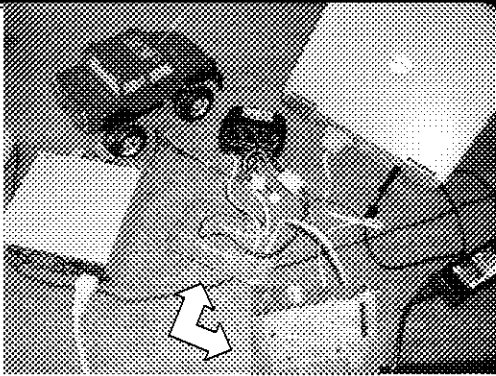


Fig.(7-a) The system with power supply

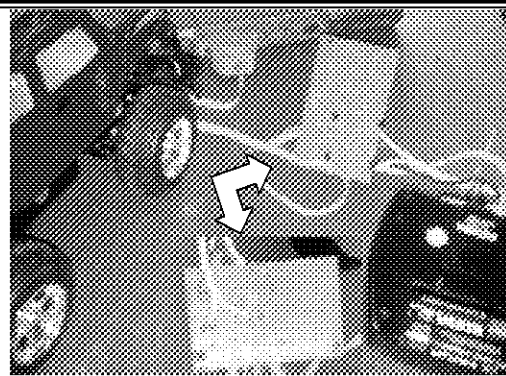


Fig.(7-b) The system without power supply

The circuit used four resistors of $2k\Omega$ for each. One is connected to one of the serial pins of the interface from one side and connected to the second part of the circuit by the other side.

The transistor that is used in this circuit is BC33740 NPN¹⁹, it is general purpose therefore it is used as a switch, transistor pins are arranged according to Fig.(8-a) and are attached as follows: B connects the serial port that receives signal, C is connected to the positive vehicle remote controller, and E is connected to the negative vehicle remote controller.



Fig.(8-a) Transistor's pins¹⁹

If the circuit is implemented as illustrated in the above section, the circuit will work properly if and only if the applied voltage from serial port is very small otherwise the transistor will explode, therefore a resistor is used to control the base current. So the proper circuit will be as shown in the Fig(8-b):

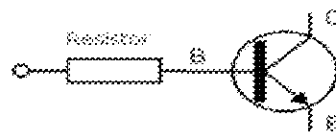


Fig.(8-b) Full circuit for one pin of the serial port¹⁹

When an AC signal amplifier is used, the transistors Base biasing voltage is applied in such a way that it always operates within its "active" region, that is the linear part of the output characteristics curves are used. However, both the NPN & PNP type bipolar transistors can be made to operate as "ON/OFF" type solid state switches by biasing the transistors base differently to that of a signal amplifier. Solid state switches are one of the main applications for the use of transistors, and transistor switches can be used for controlling high power devices such as motors, but they can also be used in digital electronics and logic gate circuits¹⁹. If the transistor used as switch then it has two modes the first

mode is OFF that realized when the volt of the serial port is low (zero) in this case, the circuit will be as in Fig.(9).

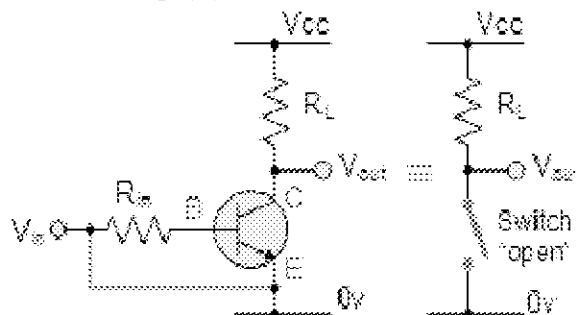


Fig.(9) Opened circuit¹⁹

The second mode of the transistor is active this case can be realized when the volt of the serial port is high (3volt), then the circuit of transistor will be as in Fig.(10).

Then the transistor operates as a switch. With a zero signal applied to the Base of the transistor it turns "OFF" acting like an open switch. With a positive signal applied to the Base of the transistor it turns "ON" acting like a closed switch.

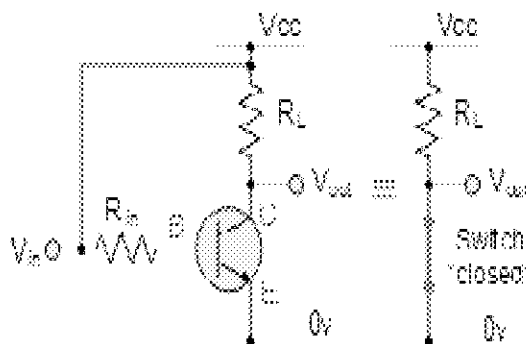


Fig.(10) Closed circuit¹⁹

7-Software Implementation

The implemented S/W will control the vehicle's movement through the computer; it will send digital signals to the designed interface through the LPT1 port, which in turn will activate the DC motors of the vehicle. The S/W contains two main parts:

The first part is the Keyboard Control: this part of the program allows the user to control the vehicle by using the keyboard keys. The arrows on the keyboard enable the user to move the vehicle to four directions; forward, backward, left and right. To stop the movement the user should press two opposite keys. Fig.(11) represents the main form, while Fig.(12) shows the key form of movements.

The second part is the Neuro-Path control: this part of S/W contains two stages: *First stage:* the user draws the path that the vehicle will move on, as the desired movement; a FFNN²⁰ shown in Fig.(13), was learned by the BP algorithm²¹ to

follow this desired movement. Fig.(14) illustrates the algorithm. Fig.(15) shows the training procedure.

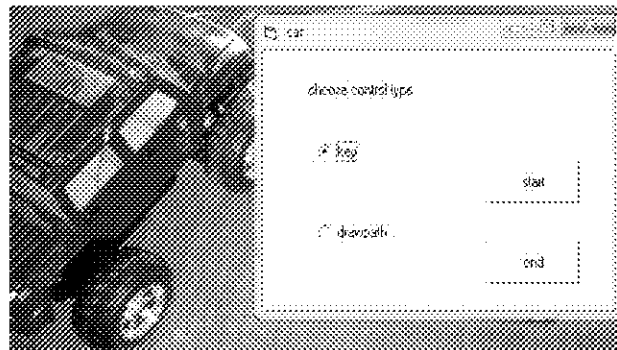


Fig.(11) Main form

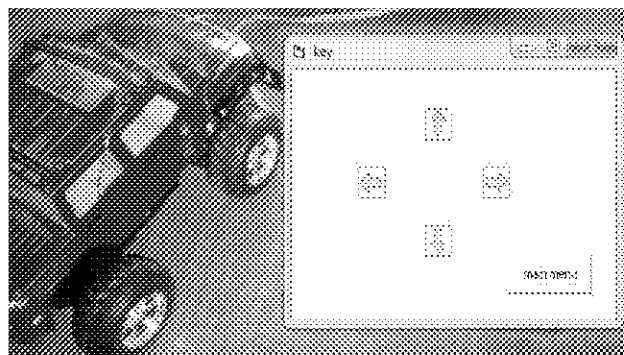


Fig.(12) Key-form

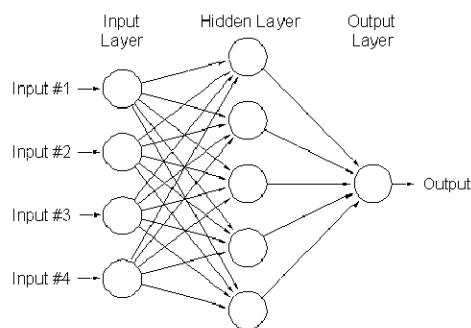


Fig.(13) Simple FFNN¹⁵

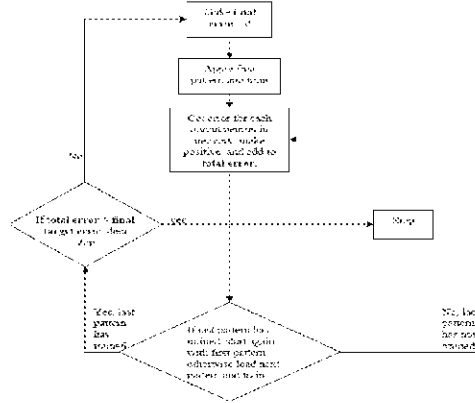


Fig.(14) BP training algorithm

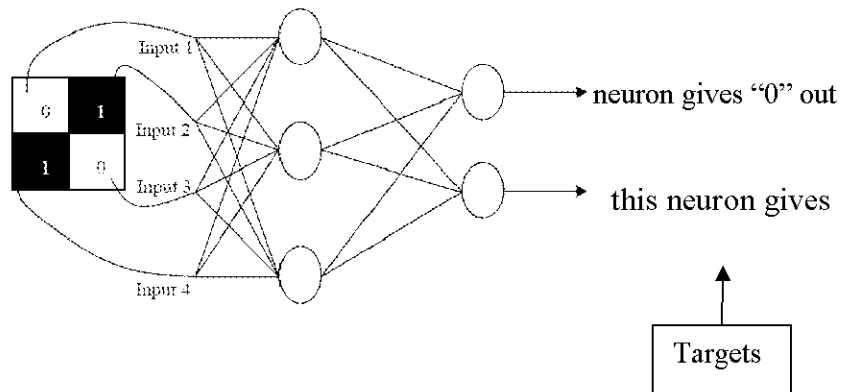


Fig.(15) applying a training pair to a network

the 1st click on the mouse will determine the starting point, while the 2nd click will draw a horizontal or vertical line between the starting point and the clicked even if it was inclined and so on, till finishing the path drawing; with the capability of moving vehicle backward with side movements. In this way the user controls is limited to draw a vertical or horizontal line since The vehicle can't move on curves. The forward lines in the path will be in black color. The reverse lines will be in red color as shown in Fig.(16).

Second stage: this stage starts when the user clicks the START button which will initiate the process of analyzing the path and the wanted movements and will send the suitable signals to execute the desired path.

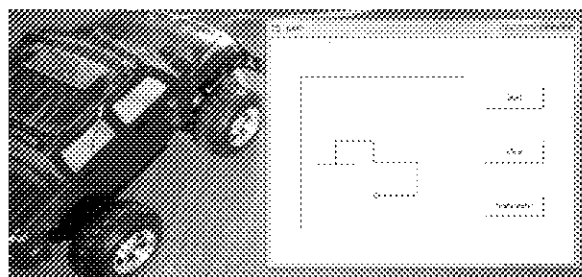


Fig.(16) Neuro-Path form

The program contain two parts of controlling, each part has distinct approach from other. The first part is key control; it can be explained as in the followed section:

in the stage of the dependence on arrow keys by using key down and key up properties. These properties will verify the pressed key(s) (one key or double) depending on key code for each arrow. The key code for each arrow is as follows; up: 37, right:38, down:39 and left:40. After reading the input from keyboard the program will produce a desired signal of motion. The graphical user interface contain four pictorial representations each one will represent one direction, when any arrow from keyboard is hit, this will show up the correspondence arrow picture on form and return back to normal case when the key is released.

The code section of this part in simplicity is to move the key code argument into a variable then dedicate the motion direction by using select case statement on this variable.

Select case stmt that has eight cases, each case was assigned a specific value, the first four cases are the four directions while the remained directions are combined for four directions; up+l, up+r, down+l, down+r.

The values assigned for first four direction (up, rght, dwn, lft) are 16, 2, 4 and 1 respectively, the cause of choosing these values is corresponding to binary values of these decimal values. The correspondence binary values are 10000, 00010, 00100 and 00001 respectively for this only one pin from LPT1 interface will be enabled.

As for the combined directions, the values will be 17, 18, 5 and 6 and the corresponding binary values are 10001, 10010, 00101 and 00110 respectively in this case two pins from LPT1 will be enabled.

The enabled pin(s) (either one pin or two pins) will operate one or two of board switches this in turn operates remote switch(s).

The second stage which was the neuro-path control, and described in Fig.(16), can be explained as followed: In this stage the vehicle will be moved on fixed path (estimated by the trained network according to the desired movement), the mouse down and mouse up events will be used.

This part will analyze the chosen point to recognize the desired movement and to show it on screen if it was forward, reverse, right or left. In this part also the required time for each movement is calculated (by knowing the length) then to pass the calculated value to mv function in order to perform specific movement according to the drawn path. The form contain picture box for drawing purposes and commands.

Each mouse down event on the picture box will draw point and line between this point and previous point, each point is x and y dimensions each dimension will be kept in an array to calculate the difference between adjacent

point then to check the result if no change on x-axis and the change will be on y-axis only and, movement will be either forward or backward, and vice versa for changing on x-axis. The line between two points represent the movement distance, this distance evaluated by multiplying the length of the line on box by an operator calculated by dividing picture box height on the range of vehicle movement (height/range), in this way the line drawn exactly from bottom to up is the range of vehicle.

When drawing path is finished, hit a command to calculate the needed information (time, direction and offset). For each calculation of distance the program will record direction signal in an array, many tests on this array will be hold then calling mv function with specific parameter according to the results of the testing operations.

These testing operations are implemented in the following algorithms:

```

Begin
D= forward
Crd=d //car direction is forward as default
If d=forward then
Begin
    If dr(k)= forward then Call mv dist(k),dr(k)
    Else if dr(k)= backward then
Begin
    Call mv dist(k),dr(k)
    D=forward // move to back but the car direction stay forward
End
Else if dr(k)= left then
begin
    Call mv dist(k),dr(k)
    D=left // car direction will be turn to left
End
Else if dr(k)=right then Begin
    Call mv dist(k),dr(k)
    D=right // car direction will be turn to right
End
End
End of the main
    
```

7.1-The training Neuro-path

The network is first initialized by setting up all its weights to be small random numbers between -1 and +1. Next, the input pattern is applied and the output is calculated (this is called the forward pass). The calculation gives an output which is completely different to what desired (the Target), since all the weights are random, the error is calculated for each neuron, which is essentially: Target – Actual Output. This error is then used mathematically to change the weights in such a way that the error will be reduced. In other words, the Output of each neuron will be closer to its Target (this is related to what is called BP²¹).

The process is repeated again and again until the error is vanished. The following explains the training procedure related to Fig.(17).

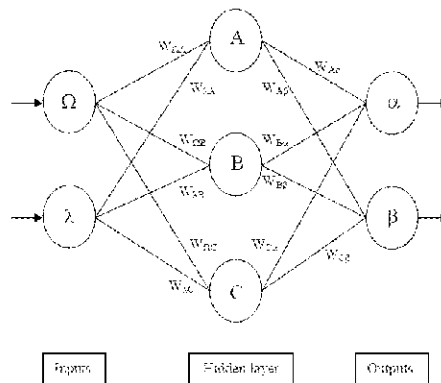


Fig.(17) Calculations of BP algorithm

1st Group of equations to calculate errors of output layer neurons:

$$\delta\alpha = out\alpha (1 - out\alpha) (Target\alpha - out\alpha)$$

$$\delta\beta = out\beta (1 - out\beta) (Target\beta - out\beta)$$

2nd Group of equations to calculate the change of output layer's neurons' weights:

$$W+A\alpha = WA\alpha + \eta\delta\alpha outA; \quad W+A\beta = WA\beta + \eta\delta\beta outA$$

$$W+B\alpha = WB\alpha + \eta\delta\alpha outB; \quad W+B\beta = WB\beta + \eta\delta\beta outB$$

$$W+C\alpha = WC\alpha + \eta\delta\alpha outC; \quad W+C\beta = WC\beta + \eta\delta\beta outC$$

3rd Group of equations to calculate errors of hidden layer neurons:

$$\delta A = outA (1 - outA) (\delta\alpha WA\alpha + \delta\beta WA\beta)$$

$$\delta B = outB (1 - outB) (\delta\alpha WB\alpha + \delta\beta WB\beta)$$

$$\delta C = outC (1 - outC) (\delta\alpha WC\alpha + \delta\beta WC\beta)$$

4th Group of equations to calculate the change of hidden layer's neurons' weights:

$$W+\lambda A = W\lambda A + \eta\delta A in\lambda; \quad W+\Omega A = W+\Omega A + \eta\delta A in\Omega$$

$$W+\lambda B = W\lambda B + \eta\delta B in\lambda; \quad W+\Omega B = W+\Omega B + \eta\delta B in\Omega$$

$$W+\lambda C = W\lambda C + \eta\delta C in\lambda; \quad W+\Omega C = W+\Omega C + \eta\delta C in\Omega$$

The constant (η) is called the learning rate to speed up or slow down the learning as required.

The algorithm to be executed in steps, the main steps are the first and second that are described as follows:

Private Sub Combo1_Click()

a = Combo1.List(Combo1.ListIndex) //left represents height

noinp = a * b

order

flagedim = True: flagew = True

End Sub

Private Sub Combo2_Click() //right represents width

b = Combo2.List(Combo2.ListIndex)

noinp = a * b

```
order
flagedim = True: flagew = True
End Sub
Public Sub order()
h = Frame1.Height / a: w = Frame1.Width / b
For i = 0 To 80
P1(i).Height = h: P1(i).Width = w: P1(i).BackColor = vbRed
Next i
topp = 100: leftt = 0: t1 = 0: t2 = b
For j = 1 To a
For i = t1 To t2
P1(i).Top = topp
P1(i).Left = leftt
leftt = leftt + w
Next i
t1 = b * j
t2 = t1 + (b - 1)
leftt = 0
topp = (h * j) + 100
Next j
End Sub
```

And the second step that is to draw line by pressing right click and moving the mouse, to clear; then press ft click and move the mouse

```
Private Sub P1_MouseDown(Index As Integer, Button As Integer, Shift As Integer, x As Single, Y As Single)
t = Button
End Sub
Private Sub P1_MouseMove(Index As Integer, Button As Integer, Shift As Integer, x As Single, Y As Single)
If t = 1 Then
P1(Index).BackColor = RGB(255, 10, 40)
Else: If t = 2 Then P1(Index).BackColor = vbRed
End If
End Sub
```

8-Conclusions and Suggestions for Future work:

During the work in this paper, a lot of problems were overcome, some of problems were solved, others are improved and many ideas would be produced as a future work. The following are the most important problems that faced the work and were solved:

1. In this paper, A progress was done in both parallel fields; S/W and H/W.
2. The movement of the vehicle was improved and the challenge of the vehicle's H/W was overcome by applying the neur-path, so the movement became in all directions.
3. The problem of using the power supply in the designed system is solved by using transistors as switches instead of a relays.

4. Rare use of parallel port used in the interface cause the major problem in this work, so it is a challenge to find suitable PC so laptop used is Dell latitude d610 1.7GHz Intel Pentium with 2GB of RAM, interfaces LPT1, running Window 7.
5. The right chose of the S/W, make the control of the vehicle movement clear and easy. This was done by using the ability of VB powerful *Graphical User Interface* (GUI), which enables a compatible design to all forms of different stages of the designed system.
6. The interface program in VB faces problem that was the definition of external procedures between the port and the program. These procedures are stored into a Dynamic Library Link (DLL) files. These files (i.e., DLL) cannot be written with VB, so this problem was solved by writing them with C⁺⁺¹⁸ language since VB can import such programs from other sources.

As a suggestion for future work, many applications would be benefit from this paper, for example:

1. To improve the algorithm, so the movement not to go in straight lines only, but to follow the edges of any given shape, i.e. to move on curved lines. This would be solved by making feedback from the vehicle and making the control as closed-loop system.
2. To fix a camera on the vehicle, so the S/W related to the used artificialNN could be improved i.e. to move according to the photograph shots taken from the fix camera.
3. To use vehicle with built in camera, then to be used into a super market or other places as a keeper that can move according to a pre-determined paths.
4. To focus on improving the control algorithm, on on-line self-learning during normal operation and on application of machine vision algorithms for visual vehicle.

References

- 1-“Mobile robot control using combined neural-fuzzy andNN”, Vukosavljev, S.A. Kukulj, D. ; Papp, I. ; Markoski, B. , Computational Intelligence and Informatics (CINTI), IEEE 12th International Conference, P. 351 – 356, 21-22 Nov. 2011.
- 2-“Model Predictive Control Method for Vehicle Platoon Control Under High-latency Wireless Communication Environment”, Hao Zhou; Romesh Saigal; Francois Dion; Li Yang, Aug. 2, 2011.
- 3-“ Intelligent Car Control for a Smart Car”, Md. NazmulHasan; S. M Didar-Al- Alam; Sikder Rezwanul Huq, Inter. Journal of Computer Applications, (0975 – 8887) vol. 14, no.3, Jan. 2011.
- 4-“Mathematical 3D Modelling and Real-Time Simulation of Tesla's Wireless Controlled Vehicle – Boat”, Vladan Vučković and Nikola Stojanović , FACTA UNIVERSITATIS (NT'S)SER.: ELEC. ENERG. vol. 24, p. 257-270, no. 2, Aug. 2011.
- 5-“ Modeling and Control of Coaxial Helicopters”, Chan Shu Ling & Ng Yuen Yan, http://www.ece.ust.hk/public/activities/fyp/fyp11/poster_LZ1-11.pdf
- 6-“ Electrical Vehicle Powered By Direct Methanol Fuel Cell”, Konrad Żurawski and Tomasz Pankowski, Warsaw University of Technology, Institute of Control and Industrial Electronic 00-662 Warsaw, Poland, 2009.

- 7- "Estimation and Control of an Automated Guided Vehicle", Suthep Butdee; Frederic Vignat; Anan Suebsomran, Control and path prediction of an automate guided vehicle. In: 9th Global Congress on Manufacturing and Management, Gold Coast, Australia, 12-14 Nov. 2008.
- 8- "Motion Planning and Coordination of Mobile Robot Behavior for Medium Scale Distributed Wireless Network Experiments", Daniel Montralio Flickinger, A thesis submitted to the faculty of The University of Utah in partial fulfillment of the requirements for the degree of Master of Science, Department of Mechanical Engineering, The University of Utah, Dec. 2007.
- 9- "Motion control of an omnidirectional mobile robot", Traineeship report, Dr.M. H. Ang Jr. and Prof. dr. ir. M. Steinbuch, Dynamics and Control Technology Group, Eindhoven, Sep. 18th, 2006.
- 10- "Estimation and Control of a Quadrotor Vehicle Using Monocular Vision and Moiré Patterns", Glenn P. Tournier; Mario Valentiy; Jonathan P. Howz; and Eric Feronx, AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado 21 - 24 Aug. 2006.
- 11- "Wireless Control of Vehicle Mirror System Using Head Movement and PIC Microcontroller", Nidal F. Shilbayeh and Mahmoud Z. Iskandarani, Journal of bComputer Science vol.1, no.2, p.182-188, 2005.
- 12- "Formations of Autonomous Vehicles Using Global Positioning Systems (GPS)", Kourosh Rahnamai; Kevin Gorman; Andrew Gray and Payman Arabshahi, <http://faculty.washington.edu/paymana/papers/aero05-2.pdf>
- 13- "New Developments and Research Trends for Intelligent Vehicles", Li Li; Jingyan Song; Fei-Yue Wang; Wolfgang Niehsen and Nan-Ning Zheng; The IEEE International Intelligent Vehicles Symposium and the IEEE International Intelligent Transportation Systems Conference, Vienna, Austria, p.10-14, 13-16 Sept. 2005.
- 14- "DSP-Based Control of Mobile Robots", Masár and M. Gerke, <http://www.prt.fernuni-hagen.de/pro/faak/Papers/EDERS04.pdf>
- 15- "Neural Networks for System Identification", Aasma Q. Shareef, A thesis submitted to the faculty of The University of Technology in partial fulfillment of the requirements for the degree of Master of Science, Dep. of Control and systems Engineering, The University of Technology, 1995.
- 16- <http://www.senet.com.au/~cpeacock> (Standard Parallel Port)
- 17- <http://en.wikipedia.org/wiki/LPT>.
- 18- <http://www.logix4u.net> (DLL libraries).
- 19- http://www.electronicstutorials.ws/transistor/tran_4.html
- 20- http://en.wikipedia.org/wiki/Neural_network
- 21- http://en.wikipedia.org/wiki/Back_propagation

سيطرة الشبكة العصبية لاسلكيا على حركة مركبة

د. أسماء قاسم شريف

جامعة بغداد/كلية العلوم/قسم علوم الحاسبات

الخلاصة:

الهدف من هذا البحث هو للسيطرة على تحركات مركبة لاسلكيا باستخدام طريقتين: إما أن تتحرك في الاتجاهات الأربعة الرئيسية من خلال لوحة المفاتيح من جهاز كمبيوتر، أو لإجبار المركبة لتتحرك ضمن المسار المطلوب من خلال تدريب الشبكة العصبية من نوع متعددة الطبقات ذات التغذية إلى الأمام، و تم ذلك تحت ثلاث مراحل: المرحلة الأولى بتصميم كيان مادي محدد ممثل في لوحة الكترونية، والذي يمثل دائرة انتقال ليكون مُعشيق مع الكمبيوتر للسيطرة على حركة المركبة لاسلكيا، في حين أن المرحلة الثانية تتم بتنفيذ برامج معينة للسيطرة على الحركة التي بُرمت باستخدام لغة اساس بصرية كائن الموجهة، والمرحلة الثالثة هي بتدريب الشبكة العصبية من نوع متعددة الطبقات ذات التغذية إلى الأمام لتخمين المسار المطلوب من حركة المركبة بواسطة خوارزمية التوليد الرجعي، ومن ثم السيطرة على حركة المركبة وفق المسار المطلوب.