

- Manipulators (mechanical structure).
- Actuators (power generation and transmission).
- Control units (electronic parts and program).
- End effects (carrier tools, hook, pencil, screwdriver, etc.).

The main part of a robot arm is the control unit. If the control unit is suitably designed, only by changing the parameters of the control program it is possible to have different jobs for the robot. In a system, before starting the target works, mathematical model of the system is simulated in a PC and may be shown on the monitor. Simulation provides investment in work power, time and material.

THE CONTROL PROBLEM

The plant considered is an arm (manipulator) with two links and two joints: a fixation joint and a joint between the two links. The movement develops in a horizontal plane. In this study we consider a control system designed to make the output of the plant track a desired angle vector $[Q1 \ Q2]$. As the main focus is on the performance of an External Model Paradigm (EMC) control using an ANN (Artificial Neural Networks), a simple feed-forward proportional controller is taken as starting point. The constants of proportionality were chosen in such a way that the generated values of torque make the first link to go to the final position before the second link. This heuristic corresponds, roughly, to the movement of human arm holding an object⁽²⁾.

TYPES OF PATH GENERATED

Industrial robots can be programmed from a distance to perform their required and preprogrammed operations with different types of paths generated through different control techniques. The three different types of paths generated are Point-to-Point Path, Controlled Path, and Continuous Path⁽³⁾,

1. ***Point-to-Point Path:*** Robots programmed and controlled in this manner are programmed to move from one discrete point to another within the robot's working envelope. In the automatic mode of operation, the exact path taken by the robot will vary slightly due to variations in velocity, joint geometric, and point spatial locations. This difference in paths is difficult to predict and therefore can create a potential safety hazard to personnel and equipment.

2. **Controlled Path:** The path or mode of movement ensures that the end of the robot's arm will follow a predictable (controlled) path and orientation as the robot travels from point to point. The robot's control system computer calculates the coordinate transformations required for this hardware management. Observations that result from this type of programming are less likely to present a hazard to personnel and equipment.
3. **Continuous Path:** A robot whose path is controlled by storing a large number or close succession of spatial points in memory during a teaching sequence is a continuous path controlled robot. During this time, and while the robot is being moved, the coordinate points in space of each axis are continually monitored on a fixed time base, e.g., 60 or more times per second, and placed into the control system's computer memory. When the robot is placed in the automatic mode of operation, the program is replayed from memory and a duplicate path is generated.

The best motor which can be used in robot is the stepper motors. Stepper motors are characterized by high torque and are capable of handling large loads with precise movements. The advantages of stepper motors over stopple AC or DC motors include:

1. No feedback requirement for position or speed control (open loop operation).
2. Noncumulative positional errors.
3. Precise electronic speed control using digital technology.
4. Compact size for driving large loads at low speeds.

With the advent of PC interface, stepper drive technology has advanced rapidly both in terms of flexibility and complexity.

2. ROBOT TO PC INTERFACING

In this paper the parallel port was used. The Parallel Port is the most commonly used port for interfacing. This port will allow the input of up to 9 bits or the output of 12 bits at any one given time, thus requiring minimal external circuitry to implement many simpler tasks. The port is composed of 4 control lines, 5 status lines and 8 data lines. The aim was to design new drivers and devices, which were compatible with each other and also backward compatible with the Standard Parallel Port (SPP).

The Parallel Port has three commonly used base addresses. These are listed in Table (1). The 3BCh base address was originally introduced used for Parallel Ports on early Video Cards.

This address then disappeared for a while, when Parallel Ports were later removed from Video Cards. They have now reappeared as an option for Parallel Ports integrated onto motherboards, upon which their configuration can be changed using BIOS. LPT1 is normally assigned base address 378h, while LPT2 is assigned 278h. 378h & 278h have always been commonly used for Parallel Ports. The lower case h denotes that it is in hexadecimal. These addresses may change from machine to machine.

Table (1): Port Addresses

ADDRESS	NOTES
3BCh - 3BFh	Used for Parallel Ports which were incorporated on to Video Cards - Doesn't support ECP addresses
378h - 37Fh	Usual Address For LPT 1
278h - 27Fh	Usual Address For LPT 2

Almost all-programming languages allow programmers to access parallel port using some library functions. For example, Borland C is providing "Inportb" and "Outportb" functions to read or write IO mapped peripherals. But the Visual Basic does not have any functions or support to access parallel port directly, but it is possible to add such capabilities to your Visual Basic application by writing a DLL(Dynamic Link Library) in VC++ (Visual C++) and calling its exported functions from VB as it has been done in this paper. VC++ provides two functions to access IO mapped peripherals, '_inp' for reading and '_outp' for writing. These functions are declared in "conio.h".

3. BASICS OF NEURAL NETWORK

Artificial Neural Network (ANN) is an artificial information processing system that has certain performance characteristics in common with biological neural network ⁽⁴⁾. A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the Brian in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interferon connection strengths, known as synaptic weights, are used to store the knowledge⁽⁵⁾. A neuron is an information-processing unit that is fundamental to the operation of a neural network. The block diagram of Figure (1) shows the model of a neuron, which form the basis for designing (Artificial) Neural Networks.

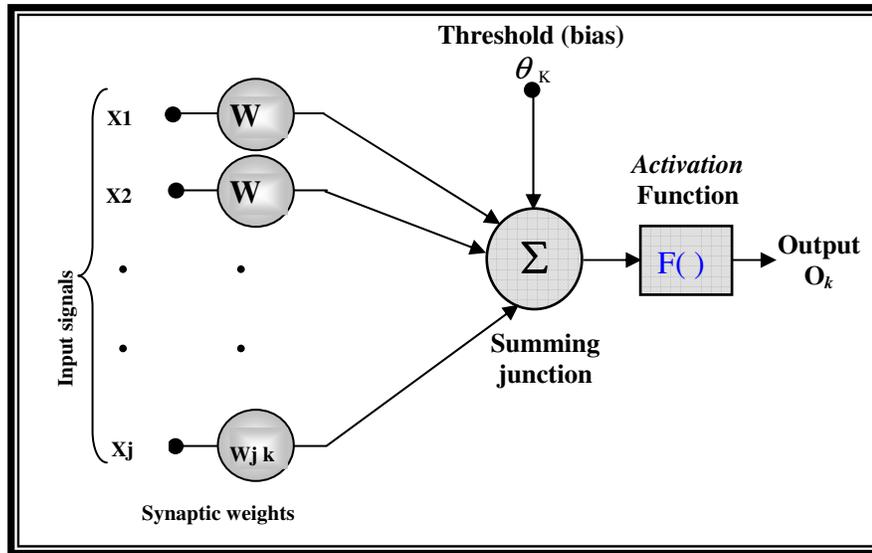


Fig. (1): Model of an Artificial Neuron

This neuron model consist of a summing junction, which adds together the weighted inputs from the other neurons, and an activation function, which generates the neuron output from the summing junction output. Also it includes an externally applied bias or threshold, denoted by (θ_k) . The bias (θ_k) has the effect of increasing or decreasing the input of the activation function $F(\)$, depending on whether it is positive or negative respectively. Neuron output signal is given by the following relationship:

$$O_k = f \left[\sum_{j=1}^n w_{jk} x_j + \theta_k \right] \quad (1)$$

Where $f(w, x, \theta)$ is the activation function.

w_{jk} : is the synaptic weight.

x_j : is the input signal.

There are several types of activation signal functions⁽⁵⁾. Figure (2) show some of the activation functions used in this work.

1. Threshold (binary step) function.
2. Hardlimiter function.
3. Threshold logic function.
4. Sigmoidal tangent function.

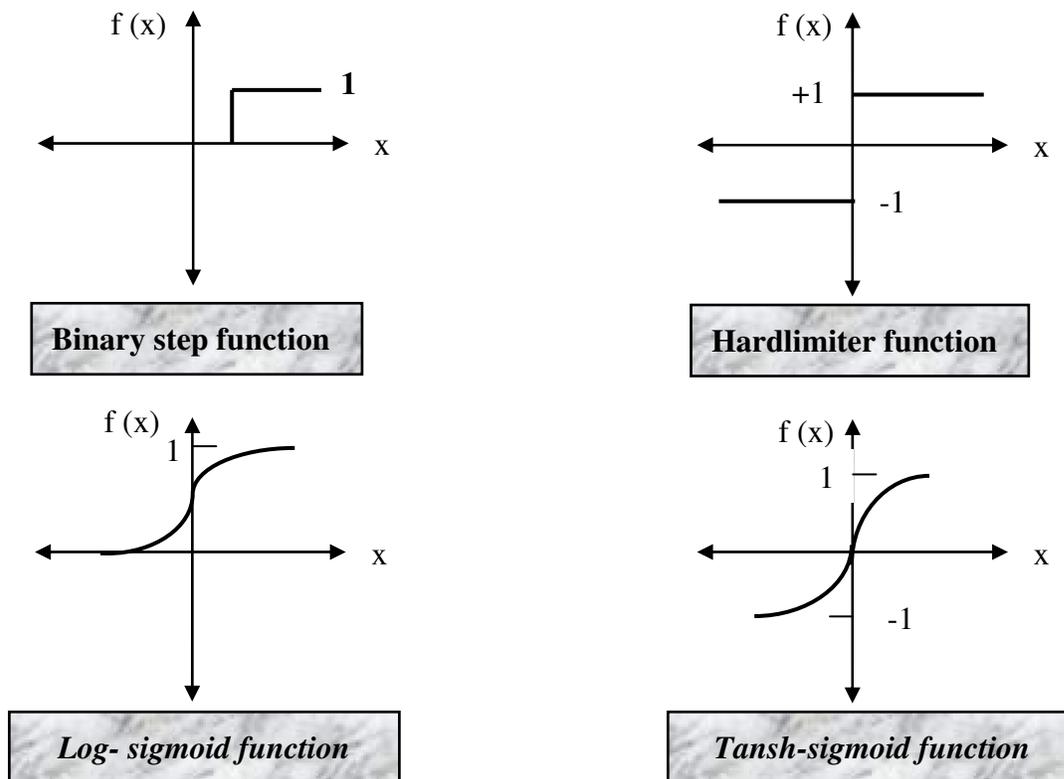


Fig. (2): Typical Neural Network Activation Function

The sigmoid function is by far the most common form of activation function used in the construction of artificial neural networks. As example of the sigmoid function is logistic function, defined by:

$$f(x) = \frac{1}{1 + \exp(-ax)} \quad (2)$$

Where (a) is the slop parameter of the sigmoid function.

4. ROBOT KINEMATICS PROBLEMS⁽⁴⁾

Focus in this section is to provide the design fundamentals for neurocontrollers used in robotics systems. Trajectory control of robotic manipulators traditionally consists of following a preprogrammed sequence of end effect movements. Robot control usually requires control signal applied at the joints of the robot while the desired trajectory, or sequence of arm end positions, is specified for the end effect. Figure (3) shows the geometry of an idealized robot manipulator with two degrees of freedom. The robot arms operate in a plane. To make the arm move, the desired

coordinates of the end effect point (X, Y) are fed to the robot controller so that it generates the joint angles (θ_1, θ_2) for the motors that move the arms. The kinematics considerations for the manipulator shown in figure (3) are based on the forward kinematics equation⁽⁴⁾. The forward kinematics equation involves mapping of joint angle coordinates (θ_1, θ_2) to the end effect position (X, Y).

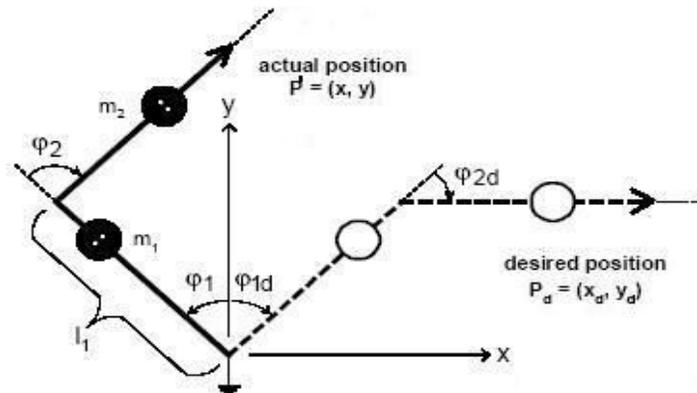


Fig. (3): Robotic Manipulator with Two Degree of freedom ($\theta=Q$): the angle between axis.

5. MECHANICAL STRUCTURE FOR ROBOT ARM

Robotic arm used for the work undertaken in this paper is a robotic arm manufactured by Feed-forward specifically for educational use. The actuation of the four joints is provided through four independent stepper motors driving a planetary gearbox and a worm gear to move each joint. The motors are not possible in this drive setup. In addition to the described drive system, the vertical axis uses a rack and pinion setup to produce linear motion. Figure (4) is a photograph of the robot used to complete the paper.

6. CONTROL PROGRAM OF ROBOT ARM

The on-line computer's capability for process control makes possible sophisticated control strategies for continuous variables without the investment in conventional digital control hardware for each loop. The objective of such control systems for continuous process variables is to apply a measured corrective action to compensate for variations from the desired level. Block diagram of the computer-aided control of the robot arm is shown in Figure (5). All the control units are gathered on only one board, which includes IC buffer, stepper motor driving and programmable I/O. Stepper motors are mounted on shoulders of the robot arms. Control program is prepared by visual basic language with its I/O VC++ DLL.



Fig. (4): Robot Arm photograph picture

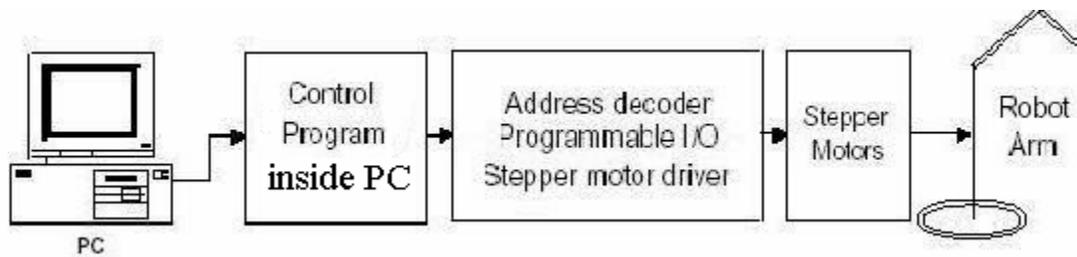


Fig. (5): Block diagram of the computer-aided control of the robot arm

The software program consists of two sections: one is the result-supported simulation program calculates the movements of the manipulator on the monitor simultaneously with the robot arm controlled. The other is the control program sending the control signals to stepper motors providing two dimensional movements of the robot arm. The Control program simulates the forward (without ANN) and inverse (with ANN) kinematics analysis of the robot arm at

initial reference position. Forward kinematics analysis provides a two dimensional movements with desired angles, directions and stepping rates depending on initial reference position. Inverse kinematics analysis provides a movement from a two dimensional position to a desired two dimensional position when the robot arms at reference position. The position of the robot arm at the two dimensional co-ordinate system is computed with the data given, and the program is done with the latest values. The result of program using two dimensional result and algorithms is shown on the monitor with (X-Y) co-ordinate system.

Control program processes and sends the program results of the variables values entered to the stepper motors of the robot arm. In the control program, the result rotation angles (Q1, Q2) are computed and the number and order of pulses to be sent the I/O interface board by considering the step angles of each stepper motors are computed. The complete design and control program was written by (VB 6 & VC++ 6) programming languages.

7. THE EMC PARALLEL INTERFACE CARD

The main purpose of this paper is to control the four main axes of Dynamation's Robot Arm by properly interfacing their Stepper Motor Driver circuit board to an IBM PC's standard parallel port. Figure (6) is a photograph of all the key points of interest on the Driver circuit board.

Key points:

- | | |
|-----------------------------------|---|
| 1: Parallel DB-25 Male Connector. | A: Dynamation Stepper Motor Driver Circuit. |
| 2: 50 Wire Ribbon Cable. | B: Buffer and line driver. |
| 3: +5VDC Power Connection. | C: Optoisolator transistor Output. |

All resistances 1000 ohm

Power Supply: A convenient source of 12v and 5v is an old IBM-PC power supply unit.

EMC Card utilizes an ULN 2003 Darlington Transistor Driver (A), Integrated circuit for communicating between the three ports (B, C) and the bus of the host computer. The base address of the Interface is 768h.

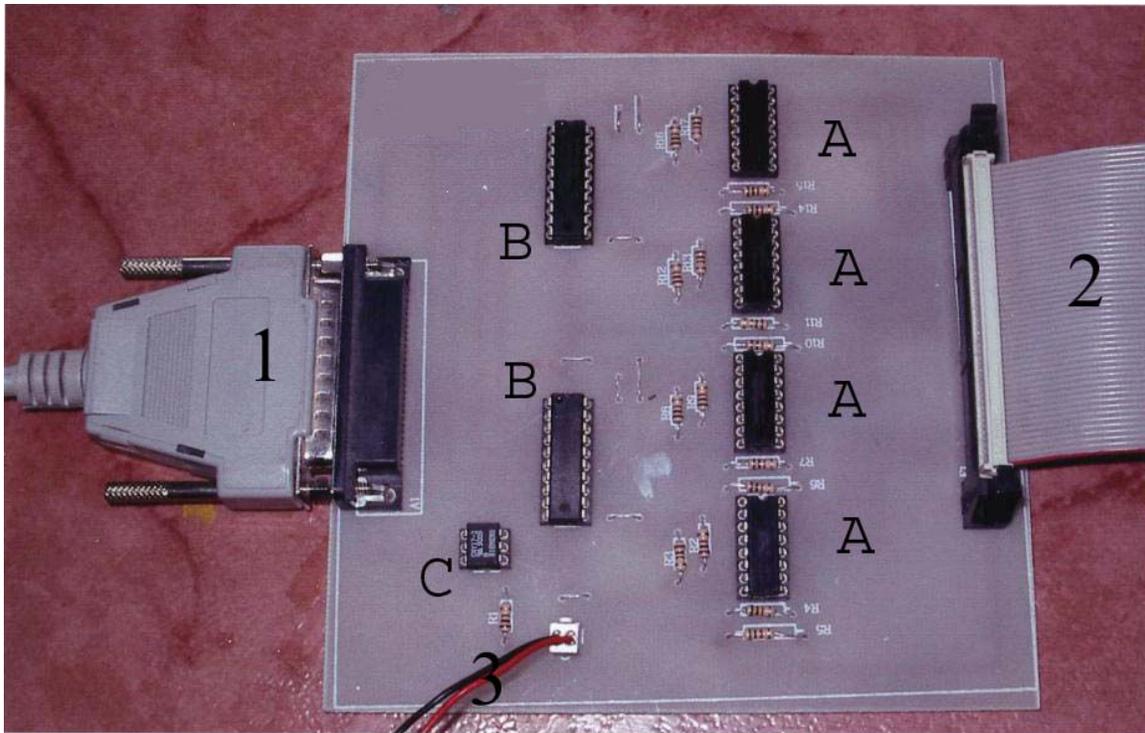


Fig. (6): External Model Circuit (EMC)

8. DYNAMATION STEPPER MOTOR DRIVER CIRCUITE (ULN2003)

The ULN2003 is high voltage, high current Darlington arrays each containing seven open collector Darlington pairs with common emitters. Each channel is rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout. This is more than adequate to control a four-phase unipolar stepper motor such as the KP4M4-00. As shown Figure (7 a). With the addition of a few resistors, you can control this kind of motor with only two data lines (compared with the Four-Wire Connection above). This uses the fact that, with Two-Coil Excitation at any time two of the coils have the inverse excitation (on/off) of the other two. As shown Figure (7 b). It is recommended to connect a 12v zener diode between the power supply and VDD (Pin 9) on the chip, to absorb reverse (or "back") EMF from the magnetic field collapsing when motor coils are switched off. You only need to supply the on/off signals for coils 3 and 2, according to the Two-Coil Excitation sequence. The corresponding inverted signals for coil (1) and the circuit itself supplies (4).

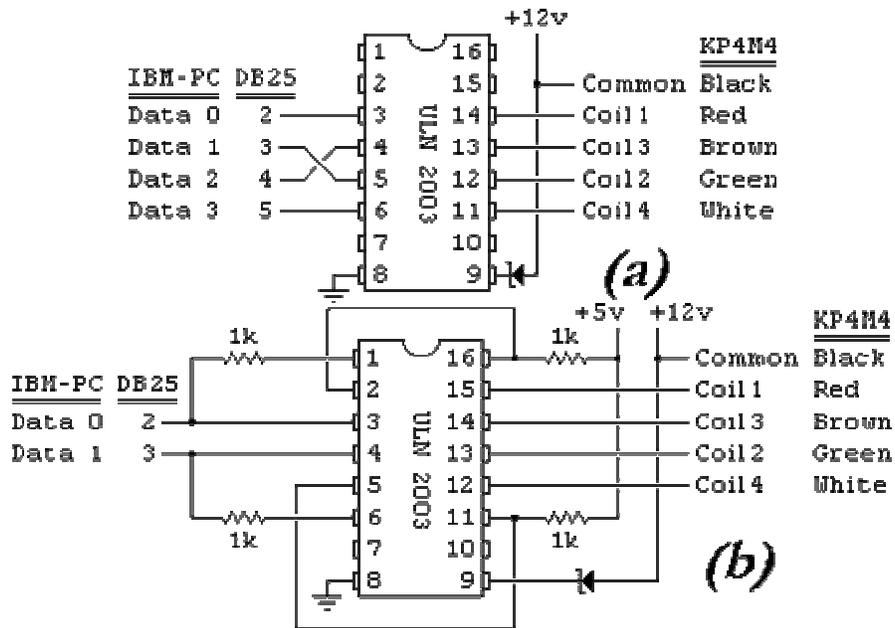


Fig. (7): Stepper Motor Driver

9. BUFFER AND LINE DRIVER (DM47LS244)

These buffers/line drivers are designed to improve both the performance and PC board density of TRI-STATE buffers/ drivers employed as memory-address drivers, clock drivers, and bus-oriented transmitters/receivers. Featuring 400 mV of hysteresis at each low current PNP data line input, they provide improved noise rejection and high fanout outputs and can be used to drive terminated lines down to 133 ohm.

Parallel Port Registers: DB-25 Parallel Connector as shown Figure (8).

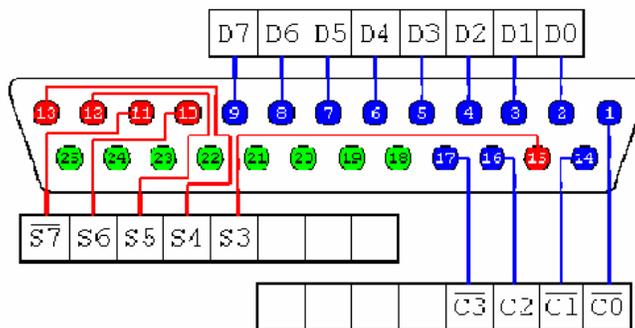


Fig. (8): Parallel Port Interface

8 Output pins: D0 to D7 => Data Register (Port Address: 0h378)

5 Input pins: S3 to S7 => Status Register

(Port Address: 0h3794 Control Register (Port Address: 0h37A)

Dynacon Test Software uses:

Data Port Register (0h378)

Status Port Register (0h379)

D0, 1 =Motor 1

S3 = Gripper Sensor feedback

D2, 3 =Motor 3

D4, 5 =Motor 4

D6, 7 =Motor 2

10. The EMC VISUAL BASIC INTERFACE PROGRAMMING

The original Visual Basic interface program, as can be seen in Figure (9), the form is a simple button of option that the user selects using the cursor keys.



Fig. (9): Main Form for the Program

Programming sequences into the robot via the host computer is also relatively complex. The method of programming sequences into the EMC controller is not very straightforward for

the beginner user. The purpose of the robot arm is to allow students to program sequences for the robot arm

11. THE FIRST: ANN METHOD FORM

To read the position of the robot after learning by using *Back propagation Algorithm*, the program must query the controller, receive input from the controller, and convert the data from the controller into a meaningful user format. In Figure (10) is a screen shot of this interface. The form contains four buttons, 'Class', 'Run', 'Clear' and 'Return'. The Class button make to see the position in the text boxes (X, Y), The run button procedure calls the Position subroutine, which queries the EMC controller and assigns the Cartesian co-ordinate position of the robot to their respective variables .The procedure then displays the co-ordinates of the robot, the angle of the gripper and the state of the gripper in the respective text boxes on the form (Q1, Q2), The Clear button was used to clear all text boxes, While the return button quits the method and return to the original program.

ANN Methode

ANN Methode

Current Robot Position

Select Class

Class 1

Class 2

Class 3

Class 4

Class 5

Input Value

X Y

After Learning

angle

Q1 Q2

Run Clear RETURN

If you Need To See how To Learn

Fig. (10): ANN Method Form.

Double click on the text “if u need to see how to learn” we can see all Back propagation Algorithm equation. Double click on the text “after learning” we can see output (Q1, Q2) without running. This program proved that the code written in the module for communicating with the controller is successful.

12. The SECOND: DIRECT METHOD

The second method is to supply the program with co-ordinates stored as positioning data for initializing the robot (such as the co-ordinates for the datum, and center points, and the locations of the ends of each conveyor). However, the difficulty is that the points are not stored in a readily recognizable format. The co-ordinates are stored as bit counts that each arm must rotate through in order to get to the desired co-ordinate, meaning, they relate directly to the angle of each arm, rather than, say, the Cartesian co-ordinates of the desired location relative to the base of the robot. As shown in the Figure (11)

Direct Method

Direct Method

Current Robot Position

Position 1 Position 2 Position 3 Position 4 Position 5

angle

Q1 Q2

0 -40

Direction

X Y

44.15 -16.0

After calculate

Run Clear New Position RETURN

If you Need Run Step By Step

Figure (11): Direct Method Form

The form contains five buttons, ‘Position’, ‘Run’, ‘New Position’, ‘Clear’, and ‘Return’. The position button make to see the position in the text boxes (Q1, Q2), The run button calculate the value (X, Y) and send data to the EMC card, New Position button use it to select new value (Q1,

Q2), the Clear button to clear all text boxes and the return button quits the method and return to the original program. Double click on the text “After calculate” we can see output (X, Y) without running. Double click on the text “if u need to run step by step” we can move the motor (1) or (2) step by step Right or Lift.

13. CONCLUSIONS

In this project, a robot arm with two degrees of freedom forward and inverse kinematics are analysed using Artificial Neural network and then according to the Movements of the actual robot arm are executed by 1.8° stepper motors controlled by a computer. Since the observation of robot movements is important before the implementation of actual system in order to prevent from possible environmental hazards, a simulation environment with control programming is prepared by Visual basic language programming with its I/O VC++ DLL.

The Project controlled systems and control program prepared is applied to a prototype robot arm and any kind of movements within the limits of arm lengths is obtained. The robot arm is a designed .to carry small load from one point to another point to have circles or arcs. For carrying the large loads, pneumatic controlled systems may be very difficult and sum calculation necessary.

In this project we found no deferent between Artificial Neural Network (ANN) and Direct Method to move the robot arm from point to point because the number of standards in the training set is small and the robot arm was kept simple (two links) in order not to mask the complexity of the problem with a non essential complexity.

REFERENCES

1. Mehmet AL-bayrak “Computer Aided Control and Simulation of Robot Arm Moving In Three Dimension”. Electrical/Computer Education Department, Isparta / Turkey, 1998 <http://w3.gazi.edu.tr/~ogurdal/IMS1998robot>
2. Darryle Wai “Graphical User Interface Development for PC-Based Control of a SCARA Robotic Arm” is available in PDF from http://www.massey.ac.nz/~wlxu/reports_pds 1997
3. Occupational Safety & Health Administration “*INDUSTRIAL ROBOTS AND ROBOT SYSTEM SAFETY* U.S. Department of Labor, <http://www.osha.gov>
4. T. Omar, A. Eskandrarian and N. Bedewi, “Vehicle Crash Modeling Using Recurrent Neural Networks”, *Mathematical and Computer Modeling* , vol.28,No.9,pp.31-24,1998.

5. D. C. reddy And K. Ghosh, "Identification and Interpretation of Manufacturing Process Patterns through Neural Networks ", Mathematical and Computer Modeling, Vol. 27, No. 5, pp. 15-30,1998.
6. D. W. patterson, "Artificial Neural Networks : Theory and Applications", Prentice-Hall, 1996
7. D. Kim, "an advanced Back propagation model for Application In Traffic Engineering" 1998.<http://blizzard.gis.uius.edu/htmldocs/Neural/neural.htm1>,

السيطرة على المفاصل المتحركة باستخدام الشبكة العصبية

مهند جابر مناتي
ماجستير هندسة حاسبات

منتظر سامي ظاهر
مدرس مساعد
كلية الهندسة-جامعة ديالى

حسين فالح مهدي
مدرس مساعد
كلية الهندسة-جامعة ديالى

الخلاصة

السيطرة على (Single Flexible Link Manipulator) الذي يحتوي على مفصلين متحركين يعتبر من المشاكل اللاخطية جدا بسبب المرونة الموزعة على ذلك المقطع. النظام الميكانيكي المستخدم في هذا الشروع هو (Robot Arm) يحتوي على مفصلين متحركين يتم السيطرة عليها بواسطة استخدام (Stepper Motor) كناقل للحركة عن طريق تصميم دائرة إلكترونية ((EMC) External Model Circuit هذه الدائرة تتكون من مجموعه من القطع الإلكترونية مثل (Buffer)، (stepper motor driver) وكذلك (programmable Input/Output). هذا النظام تم السيطرة عليه باستخدام نظريتين. النظرية الأولى هو استخدام شبكات الخلايا العصبية (Artificial Neural Networks (ANN) هذه الشبكة يتم تدريبها باستخدام نظرية (Back-Propagation). أما النظرية الثانية هو استخدام الطريقة المباشرة (direct method) أي يتم إدخال القيم التي يراد الوصول إليها بصورة مباشرة.