# PC-Based High-Speed Model Reference Motor Controller

M.Sc. Ammar A. Sohrab
Electrical Engineering Department
Al-Mustansiriya University
Iraq-Baghdad

M.Sc. Emad A. Hussein
Electrical Engineering Department
Al-Mustansiriya University
Iraq-Baghdad

## Abstract

*Personal Computer _Based Input / Output (PC-Based I/O) Interface products became increasingly reliable, accurate, flexible, simple, cost-effective, and affordable. So the PC - Based control systems are widely used in industrial and laboratory applications for monitoring, control, data acquisition, and automated testing. The PCI-bus will allow a single add-in card hardware design to be created for multiple platforms. Also the bus standards provide the bandwidth required for many new high-performance applications. The Applied Micro Circuit Corporation (AMCC) S5933 provides a flexible and low-cost, compliant interface to the PCI-bus. Software will play a vital role in developing data acquisition and control systems. The system software consists of device drivers to control the hardware, acquisition, control, and analysis. A Model Reference controller using Neural Networks (NN) will be the controller software that will be used in this application. A neural network will be trained to control an external plant so that it follows a reference model. The neural network plant model is used to assist in the controller training with better system response to a sudden change in the motor speed. The model reference architecture requires a separate neural network controller be trained off-line, in addition to the neural network plant model.*

*Key words : NN, Model Reference controller, PC-Based, AMCC S5933*

**الخلاصـــــة**

*مُنتَجات التوصيل ادخال / اخراج المعتمدة على الحاسوب الشخصي أصبحتْ موثوقة، دقيقة، مرنة، بسيطة، مربحة ورخيصة جداً. لذا فان أنظمة السيطرةِ المعتمدة على الحاسوب الشخصي كثيرة الإستعمال في التطبيقات الصناعيةِ والمختبريةِ للمراقبة، السيطرة، جمع بيانات والإختبار الآلي. الناقل PCI سَيسْمحُ لكارت واحد اضافي لتصميمِ أجهزةِ البطاقةِ الّتي تصنع لعدة برامج. أيضاً ثوابت النواقل تُزوّدُ عرض موجة مَطلوب للعديد مِنْ التطبيقاتِ الجديدةِ العالية الأداءِ. القطعة AMCC S5933 تزوّدُ التوصيل المتكامل إلى الناقل PCI مرن ورخيص. البرامجيات سَتَلْعبُ دورا حيويا في تَطوير جمع البيانات وانظمة السيطرة. برامج النظامَ تتكون من سواقِات الأداةِ للسَيْطرَة على الأجهزة، الإكتساب، السيطرة والتحليل. جهاز سيطرة مرجعي باستخدام الشبكات العصبية (NN) سَتَكُونُ برامجَ السيطرة والتي سَتَكُونُ مستعملة في هذا التطبيق. الشبكة عصبية سوف تدرب للسَيْطرَة على مشروع خارجي لذلك ستتبع نموذج مرجعي. مشروع نموذج الشبكة العصبية يُستَعملُ للمُسَاعَدَة في تدريبِ المسيطرمع استجابة افضل للنظام لتغير مفاجئ في سرعةِ المحرّكِ. معمارية النموذج المرجعي تَتطلّبُ جهازَ شبكة عصبيةِ مسيطر منفصل يدرب مستقلا، بالأضافة إلى مشروع نموذجَ الشبكة العصبية.*

## 1. Introduction

High-speed data-acquisition and control system depends on the processor speed as well as on the bus speed [1]. As shown in Fig.(1), the PCI-bus concept was developed to break the PC data I/O bottleneck and opens the door to increase system speed and expansion capabilities so that the bus place high-speed peripherals. This speed must be closer enough to the system's processor bus providing faster data transfer between the peripheral and the processor. The PCI-bus represents a microprocessor independent bus offering performance more than adequate for the most demanding applications, such as full-motion video. The single-chip solution provides a high-performance interface between PCI-bus and custom add-in cards. Address decoding and sourcing, burst transfers, and all of the logic circuits needed to perform efficient data transfers are integrated into the device [5] [6] [8]. Applied Micro Circuits Corporation (AMCC) is the premier supplier of single-chip solutions. This developed and produced the S5930-33 master/slave controller interface to solve the problem of interfacing applications to PCI-bus. A bi-directional 32-bit wide FIFO buffer included on the chip to facilitate the system-to-system synchronization and data transfers between PCI-bus and peripheral board [4].

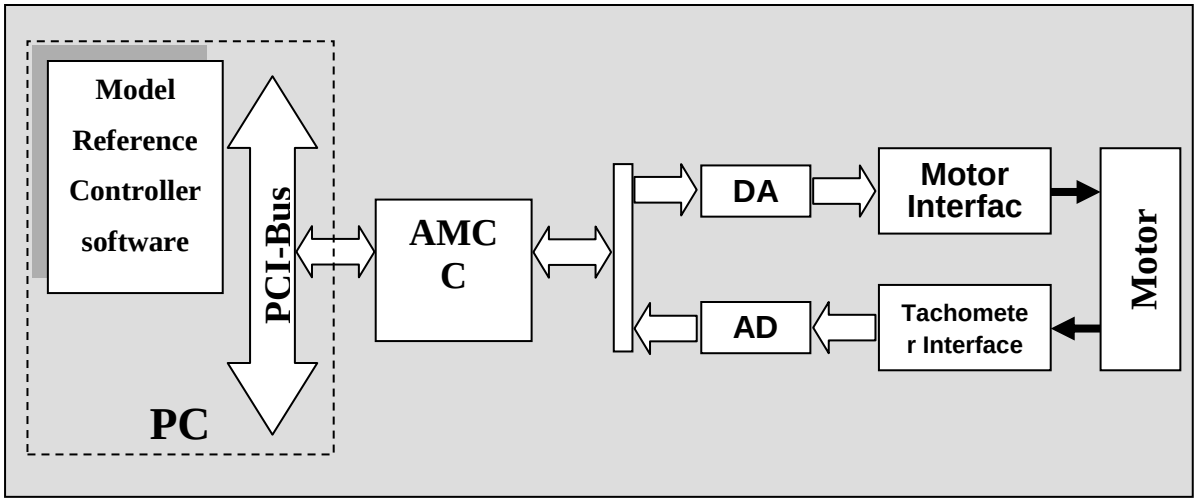


***Fig.(1): Block Diagram for High-Speed Data Acquisition System***

The software needs for data acquisition and control systems are affected by many factors, this include application requirements, computer hardware, operating systems, and the data acquisition and control hardware. The selected software must be versatile enough to accommodate divers computer architectures and data acquisition devices [1] [5] [7]. So the Model Reference Controller using Neural Network will be the selected software [9] [10]. The neural model reference control architecture (shown in Fig.2) uses two neural networks: a controller network and a plant model network [11].

The plant model is identified first, and then the controller is trained so that the external plant output follows the reference model output. Note that the external plant will be the motor system that connected with the PC.

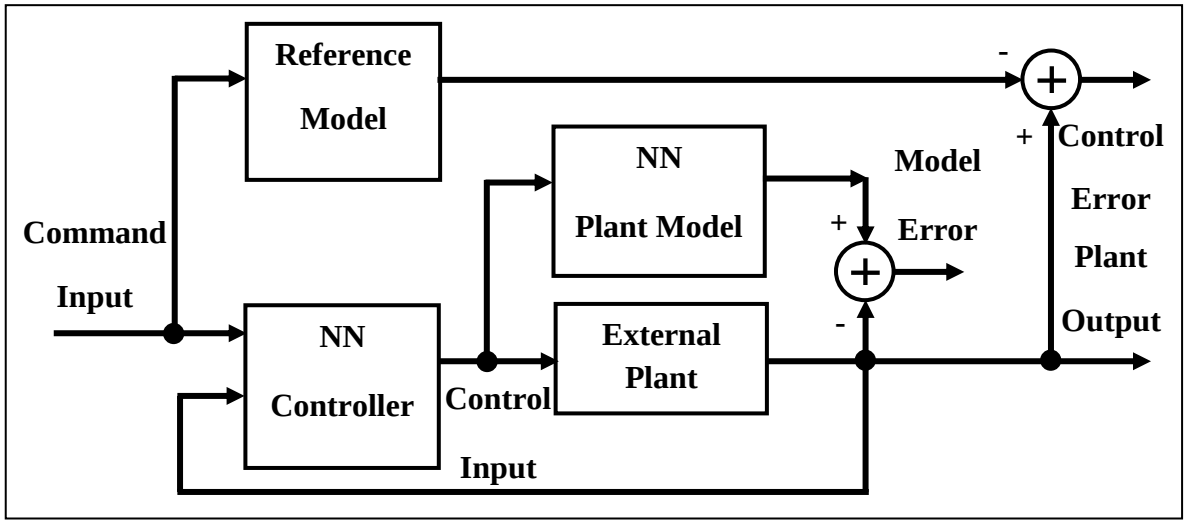


***Fig.(2): Model Reference Controller using Neural Network***

Developing and programming this system in real-time required a flexible and reliable programming language, like MATLAB. The MATLAB programming language has a practically rich set of operators and ideal for configuring programmable I/O devices and flag testing. Also it is more attractive and digestible in presenting the information in "byte" sized packet. [12] [13]

## 2. System Design

As shown in Fig. (1) the system consists of four main parts: PCI-bus, AMCC S5933, Local Bus, and Peripheral I/O system. The PCI-bus produced by Intel Corporation to meet all the requirements for data acquisition and control systems. The PCI-bus is different from older standard buses by the following points: -

1. Its functions with either a 32-bit or a 64-bit data bus and full 32-bit address bus.
2. The address and data buses are multiplexed to reduce the size of the edge connector and without any effect on bus speed.
3. The 32-bit synchronous bus which can provide data rates from 132 Mbytes/second (33 MHz) to 264 Mbytes/second (66 MHz), and doubles the throughput when using 64-bit data.

4. PCI-bus can performed all read and write transfers as single-byte or burst transfers mode.
5. Processor independent, which provides plug and play capability.
6. Allows manufacturers to significantly trim development costs by not having to completely redesign every product cycle.
7. It removes systems designers from processor treadmill by isolating the I/O subsystem from the processor / memory / cache subsystem.
8. Low latency path to memory, high performance, upgradability, and auto-configuration.

The PCI-bus can be populated with adapters requiring fast accesses to each other and for system memory and that can be accessed by the processor at speeds approaching that of the processor's full native bus speed. When the PCI-bus performs a burst mode then the length of the burst is determined by the bus master. The target is given the start address and the transaction type. As the master become ready to transfer each data item, it informs the target whether or not it is the last one. The transaction completes when the final data item has been transferred [5] [6] [8]. The block diagram in Fig. (3) shows the major functional elements within the AMCC S9533 MatchMaker. The S5933 allows special direct data accessing and movement between the PCI-bus and the user application through mailbox registers or the FIFO data channel, or a user can define and enable one or more of the four Pass-Thru data channels. Each data channel of the four Pass-Thru data channels is implemented by defining a Host memory segment size and 8/16/32-bit user bus width. The addition of two 32 byte FIFOs, also used in S5933 Bus Mastering applications, provides further versatility to data transfer capabilities. Four 32-bit Mailbox Registers coupled with a Status Register and extensive interrupt capabilities provide flexible user command or message transfers between the two buses [4].

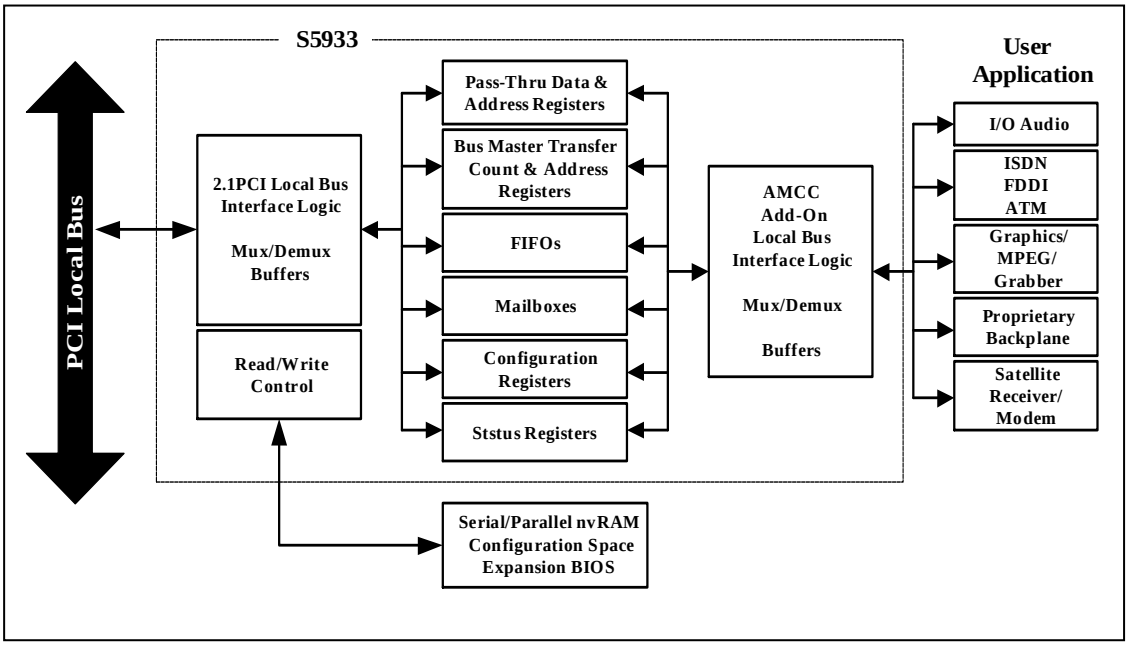


***Fig.(3): The S5933 PCI-Controller between PCI-Bus and ADD-ON local Buses***

S5933 Bus Master Direct Memory Access (DMA) data transfers to and from the PCI-bus are performed through the First Input First Output (FIFO) data channel under either Host or Add-On software control or Add-On hardware control using dedicated S5933 signal pins. Also FIFO DMA transfers are supported using Address and Transfer Count Registers. In addition, the S5933 also allows use of an external serial or byte-wide non-volatile memory to perform any pre-boot initialization requirements and to provide custom expansion BIOS or POST code capability [4]. The control and configuration of the Add-On Local bus, and the MatchMaker itself, is performed through three primary groups of registers. These groups consist of PCI Configuration Registers, PCI Operation Registers and Add-On Operation Registers. All these registers are user configurable through their associated bus or from an external non-volatile memory device.

Also the S5933 generates data and control signals. These signals are used by the local bus and then by the peripheral components. The local bus is used for maximum system expandability and for more or less demanding peripherals [4]. The I/O system consists of Analog input, and Digital Output systems (see Fig. (1)). Fig. (4) shows the schematic diagram of AD7859 that is used for peripheral equipment. For analog input system AD7859 analog-to-digital converter is used. The AD7859 is high-speed, low power, 8-channel, 12-bit ADC which operates from a single 3 V or 5 V power supply. The part requires an external 4 MHz master clock (Clock In (CLKIN)), two CREF capacitors, a (Convert Start ($\overline{\text{CONVST}}$)) signal to start conversion and power supply decoupling capacitors. The part provides the user with track / hold, on-chip reference, calibration features, A/D converter and parallel interface logic functions on a single chip. The Analog to Digital Converter (ADC) contains self-calibration and system calibration options to ensure accurate operation over time and temperature and have a number of power-down options for low power applications [3].

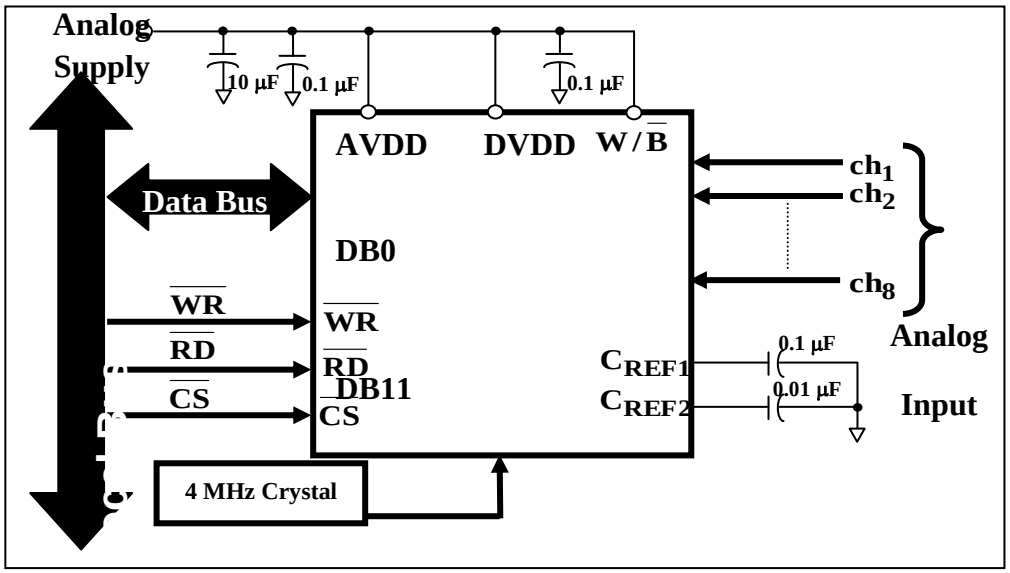


***Fig. (4) The Schematic Diagram of AD7859***

The AD7859 is capable of 200 kHz throughput rate while the input track-and –hold acquires a single in 500ns and features a pseudo-differential sampling scheme. The A/D (Analog to Digital) converter section of the AD7859 consists of a conventional successive-approximation converter based around a capacitor DAC. The AD7859 input voltage range is 0 to (Vreference) VREF (unipolar) and –VREF / 2 to + VREF / 2 about VREF / 2 (bipolar) with both straight binary and 2s complement output coding respectively. The reference input to the part is connected via a 150 kΩ resistor to the internal 2.5 V reference and to the on-chip buffer. A major advantage of the AD7859 is that a conversion can be initiated in software, as well as by applying a signal to the $\overline{\text{CONVST}}$ pin. The AD7859 can operate at throughput rates of over 200 kSPS. Another advantage of the AD7859 is that most of the control, status, output, and test signals are generated by software without hardware handling. The AD7859 provides parallel interface capable of operating in either word or byte mode. This feature can be obtained through $\text{W} / \overline{\text{B}}$ pin, which is high for word operation and low for byte operation. In this paper the word mode operation is used by connecting $\text{W} / \overline{\text{B}}$ pin to the supply voltage [3].

For analog output system the AD8582 complete dual 12-bit DAC is used. The AD8582 is a complete, parallel input, dual 12-bit, voltage output digital-to-analog converter. Only one +5 V power supply is necessary for operation. This power supply is built using a CBCMOS process, this monolithic DAC offers the user low cost, and ease-of-use in +5 Volt only systems. It contains two voltage-switched, 12-bit, laser-trimmed digital-to-analog converters, a curvature-corrected bandgap reference, rail-to-rail output op amps, input registers, and DAC registers. The reference (VREF) is trimmed to 2.5 Volts output, and the on-chip amplifier gains up the DAC output to 4.095 volts full scale. Fig. (5) shows the schematic diagram of AD8582 connected to the local bus. The parallel data interface consists of twelve data bits, DB0-DB11, an address select pin $\overline{\text{A}}$ / B, two load Strobe pins Load Port A (($\overline{\text{LDA}}$, $\overline{\text{LDB}}$)) and an active low (Chip Select ($\overline{\text{CS}}$)) strobe. In addition an asynchronous $\overline{\text{RST}}$ pin will set all DAC register bits to zero causing the VOUT become zero Volts, or to midscale for trimming applications when the MSB pin is programmed to logic 1. This function is useful for power on reset or system failure recovery to a known state [2].

The high-speed parallel data interface connects to the fastest processors without wait states. The double-buffered input structure allows the user to load the input registers one at a time, then a single load strobe tied to both LDA+LDB inputs will update both DAC outputs simultaneously. LDA and LDB can also be activated independently to immediately update their respective DAC registers. An address input decodes DAC A or DAC B when the chip select $\overline{\text{CS}}$ input is strobed. The MSB can be used to establish a preset to midscale when the reset input is strobed [2].

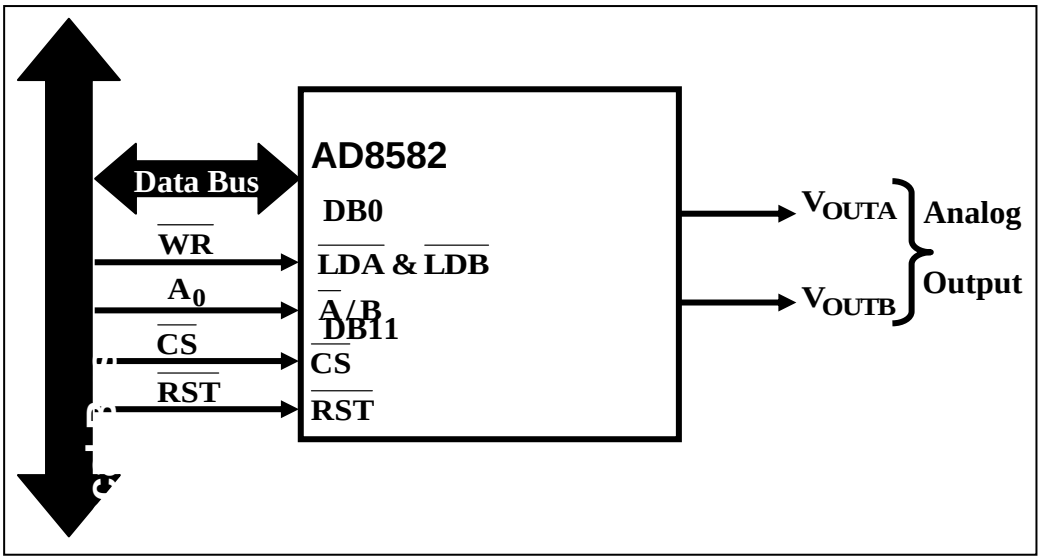


***Fig. (5) The Schematic Diagram of AD8582***

## 3. Model Reference Control

The neural model reference control architecture uses two neural networks: a controller network and a plant model network, as shown in the following figure. The plant model is identified first, and then the controller is trained so that the plant output follows the reference model output. [14] [15]

Figure (6) shows the details of the neural network plant model and the neural network controller, as they are implemented in the Neural Network Toolbox [12]. Each network has two layers, and you can select the number of neurons to use in the hidden layers. There are three sets of controller inputs:

- Delayed reference inputs
- Delayed controller outputs
- Delayed plant outputs

For each of these inputs, you can select the number of delayed values to use. Typically, the number of delays increases with the order of the plant. There are two sets of inputs to the neural network plant model: [16] [17]

- Delayed controller outputs
- Delayed plant outputs

As with the controller, you can set the number of delays. The next section demonstrates how you can set the parameters.

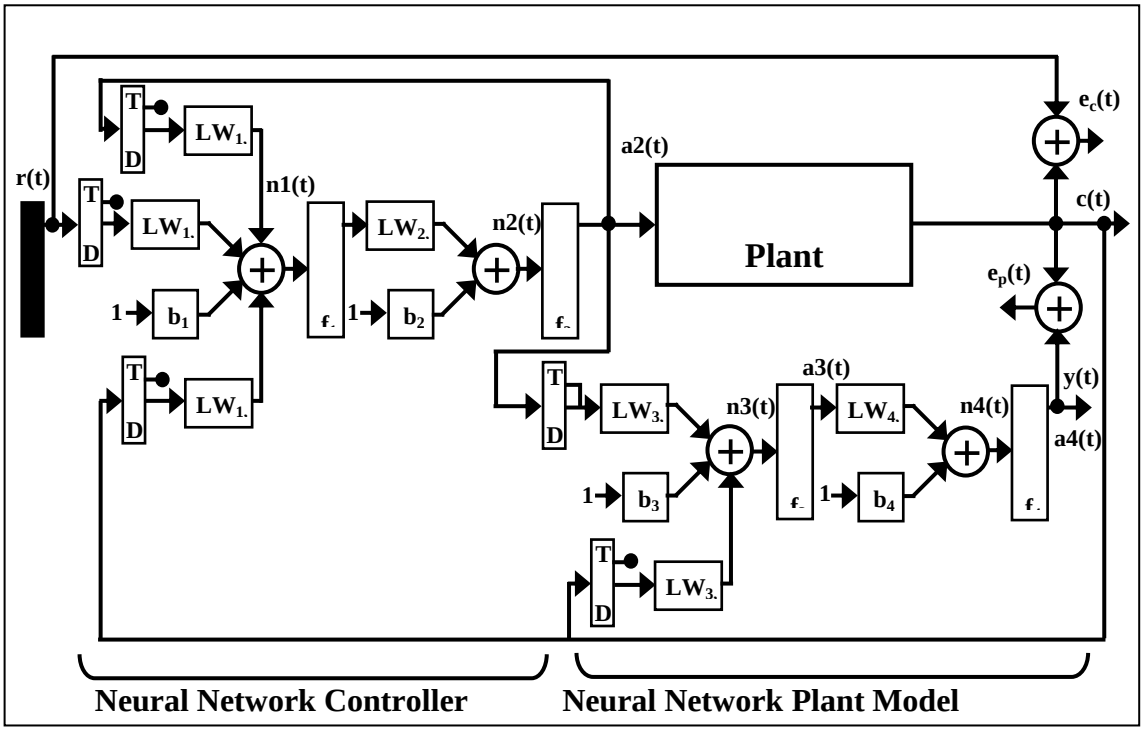


***Fig. (6) : Neural Network plant model and the Neural Network controller***

## 4. Using the Model Reference Controller Block

This section demonstrates how the neural network controller is trained. The first step is to copy the Model Reference Control block from the Neural Network Toolbox blockset to our model window.

A demo model is provided with the Neural Network Toolbox to demonstrate the model reference controller.

To run this demo, follow these steps:

**1.** Start MATLAB Simulink and creates the model window shown in Fig.(7). The Model Reference Control block has already been placed in the model.

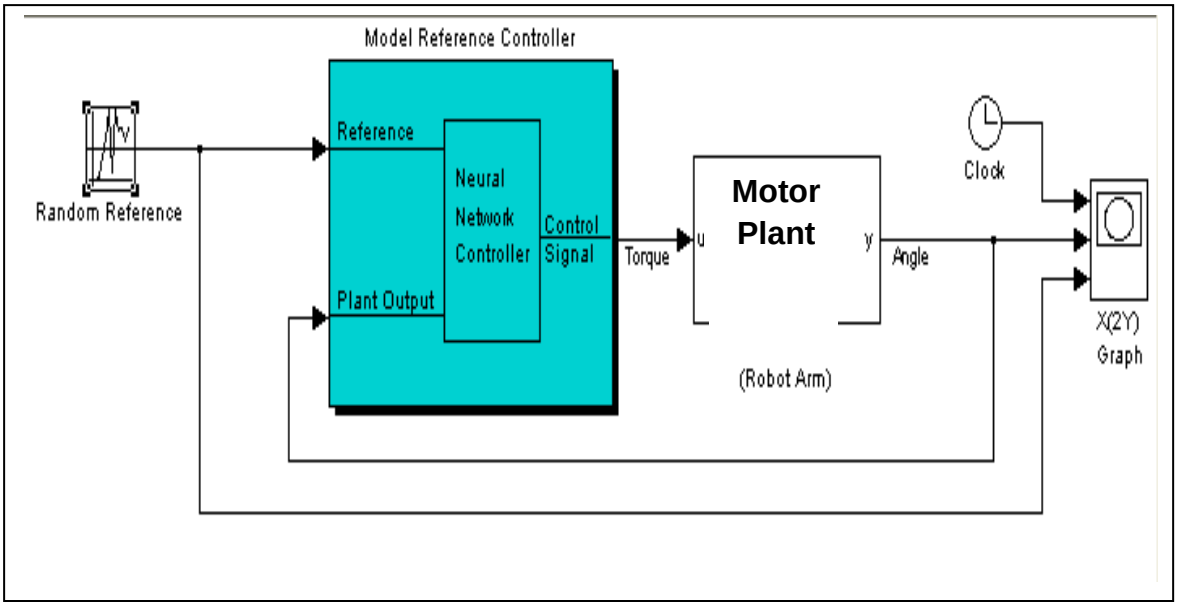


***Fig. (7) : Circuit description using Matlab simulink***

**2.** Double-click on the Model Reference Control block. This brings up the window shown in Fig.(8) for training the model reference controller.

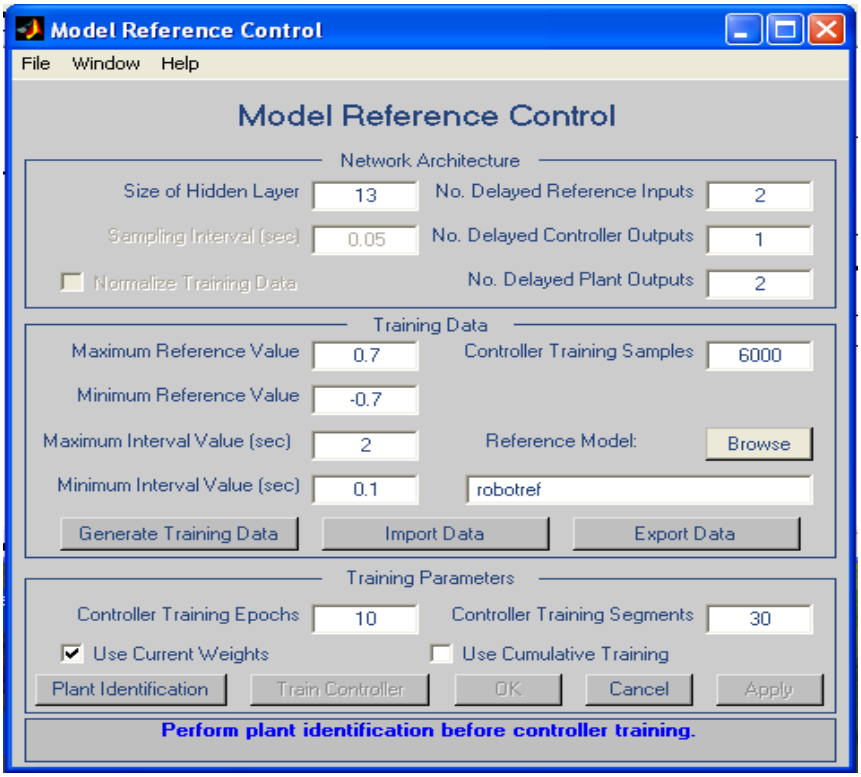


***Fig. (8) : The training module for the model reference control***

**3.** The next step would normally be to select Plant Identification, which opens the Plant Identification window. You would then train the plant model. Since the Plant Identification window is identical to the one used with the previous controllers, we won't go through that process here.

**4.** Select Generate Data. The program then starts generating the data for training the controller. After the data is generated, the window shown in Fig.(9).

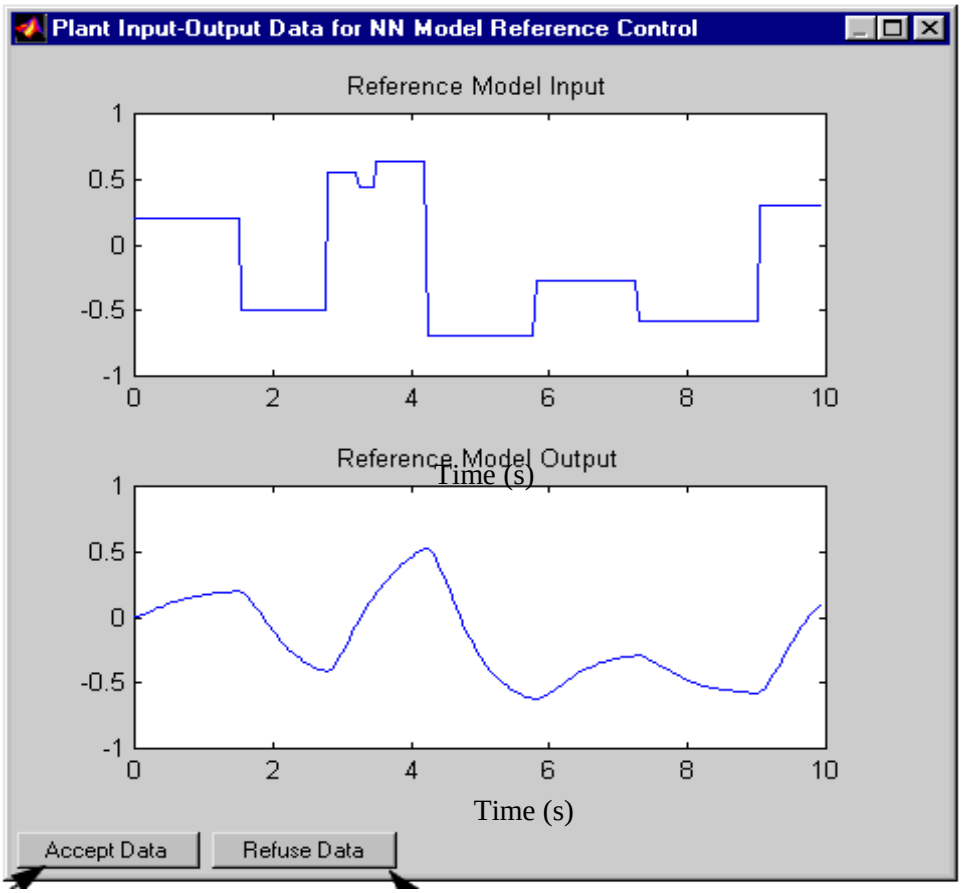


***Fig. (9) : Plant Input-Output data for NN Model***

**5.** Select Accept Data. Return to the Model Reference Control window and select Train Controller. The program presents one segment of data to the network and trains the network for a specified number of iterations (five in this case). This process continues one segment at a time until the entire training set has been presented to the network. Controller training can be significantly more time consuming than plant model training. This is because the controller must be trained using dynamic backpropagation. After the training is complete, the response of the resulting closed loop system is displayed, as in the following figure (Fig.(10)).

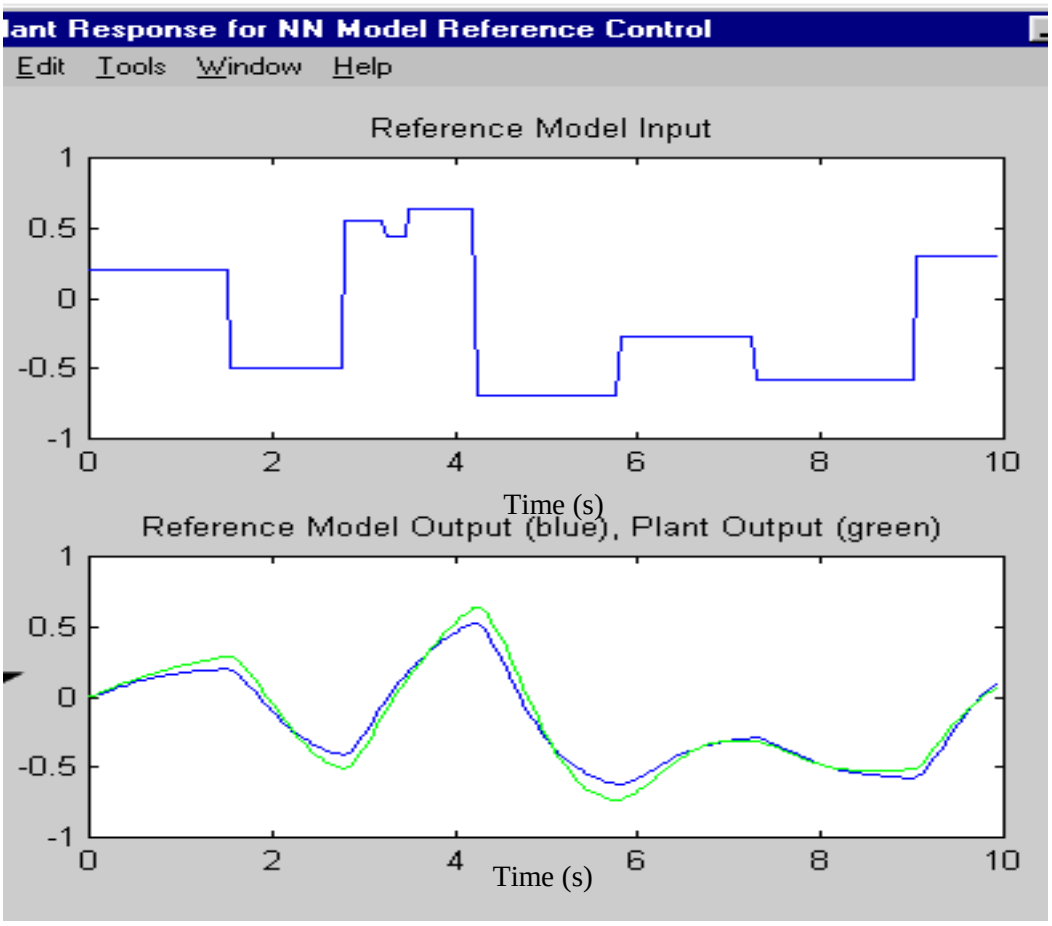


***Fig. (10) : Response for NN Model Reference Control***

**6.** Go back to the Model Reference Control window. If the performance of the controller is not accurate, then you can select Train Controller again, which continues the controller training with the same data set. If you would like to use a new data set to continue training, the select Generate Data or Import Data before you select Train Controller. (Be sure that Use Current Weights is selected, if you want to continue training with the same weights.) It may also be necessary to retrain the plant model. If the plant model is not accurate, it can affect the controller training. For this demonstration, the controller should be accurate enough, so select OK. This loads the controller weights into the Simulink model.

**7.** Return to the Simulink model and start the simulation by selecting the Start command from the Simulation menu. As the simulation runs, the plant output and the reference signal are displayed, as in Fig.(11).

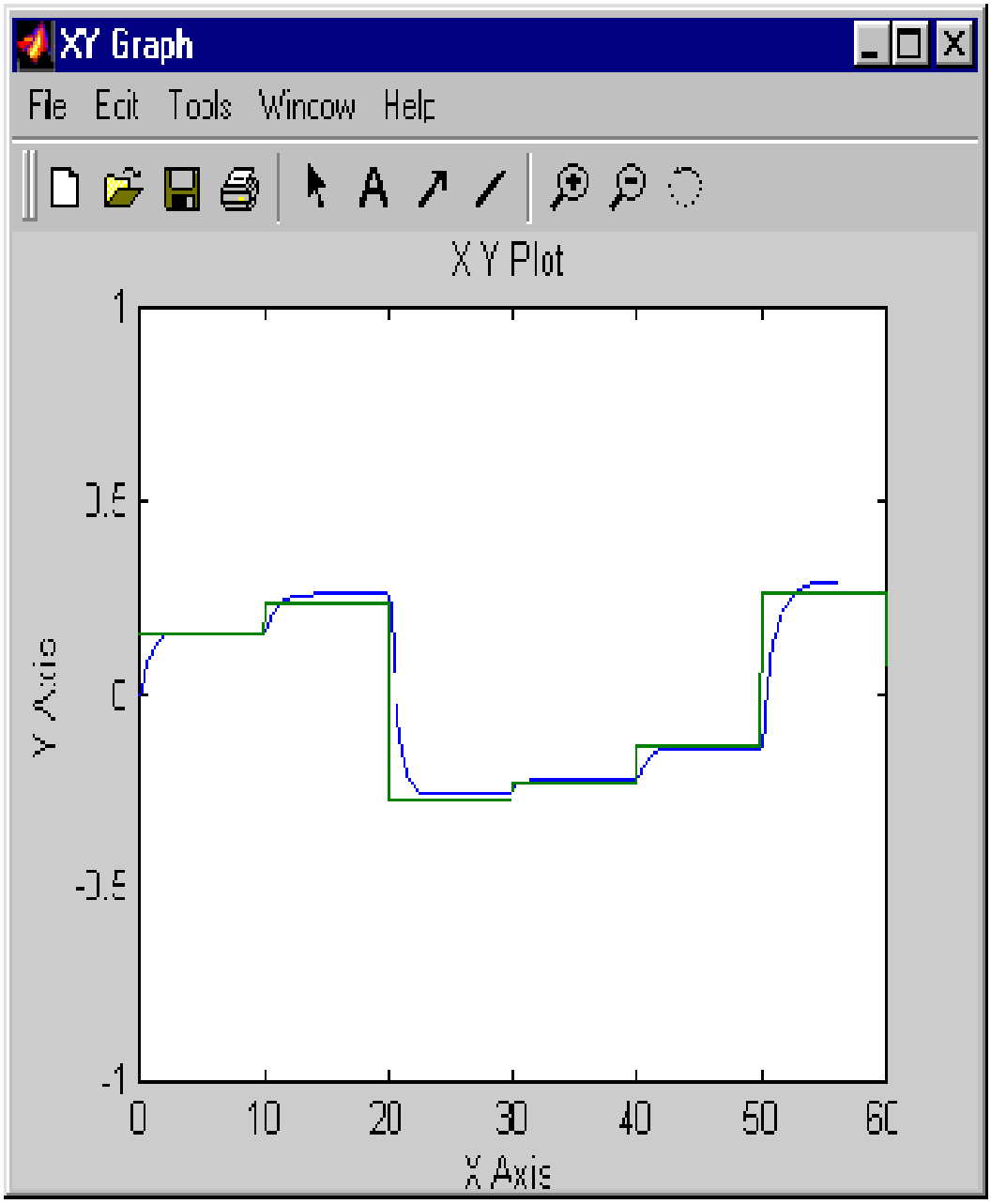


***Fig. (11) : XY Graph after simulation***

## References


**[1]** ***Advantech Corporation*****, "Total Solution For PC-Based Automation",** Advantech Corp. Press, Vol. 91, 1999.

**[2]** ***Analog Devices Corporation*****, "+5 Volt, Parallel Input Complete Dual 12-Bit DAC AD8582",** Analog Devices Corp. Press, 1996.

**[3]** ***Analog Devices Corporation*****, "3V To 5V Single Supply, 200 KSPS, 8-Channel, 12-Bit Sampling ADCs AD7859 / AD7859L",** Analog Devices Corp. Press, 1996.

**[4]** ***Applied Micro Circuits Corporation*****, "S5933 32-Bit PCI MatchMaker",** Applied Micro Circuits Corp. Press, October 1998.

**[5]** ***Barry B. Bery*****, "The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, and Pentium Pro Processor Architecture, Programming, and Interfacing",** Prentice-Hall, 1997.

**[6]** ***Guy W. Kendall*****, "Inside the PCI Local Bus",** Byte, Vol. 19, No. 2, February 1994.

**[7]** ***Howard Hutchings,*** **"Interfacing With C",** Electronics World + Wireless World, Part 1-4, April-July 1990.

**[8]** ***Richard Nass*****, "PCI-Bus Links Directly To Processor Bus",** Electronic Design, Vol. 42, No. 12, June 13, 1994.

**[9]** ***Hagan, M. T., H. B. Demuth, and M. H. Beale*****, "Neural Network Design",** Boston, MA: PWS Publishing, 1996.

**[10]** ***D. W Patterson*****, "Artificial Neural Networks, Theory and Applications",** Prentice Hall, Simson and Schuster (Asia) Pte. Ltd., 1996.

**[11]** ***F. K. Rasheed*****, "Analysis of Fuzzy Model Reference Learning Controller",** M.Sc. Dissertation, Electrical Engineering Dept., Al-Mustansiryah University, November. 2000.

**[12]** ***H. Demuth, & M. Beal*****, " Neural Network Toolbox for Use With MATLAB",** User's Guide, Mathworks, Inc., Version 4.0,7th printing. 2002. Web Http:// www. MathWorks. Com.

**[13]** ***J. A. Freeman, & D. M. Skapura*****, " Neural Networks: Algorithms, Applications, and Programming Techniques",** Addison-Wesley Publishing Company, Inc.,2001.

**[14]** ***E. J. Rzempoluk*****, "Neural Network Data Analysis Using SimulNETTM",** Springer-Verlag New York, Inc., 1998.

**[15]** ***Y. Ichicakawa*****, " Neural Network Application for Direct Feedback Controllers",** Member, IEEE, and Toshiyuki Sawa, 1992.

**[16]** ***L. C. Jain & R. K. Jain*****, "Hybrid Intelligent Engineering systems",** World Scientific Publishing, Co. Pte. Ltd., 1997.

**[17]** ***F. C. Chen*****, " Back-Propagation Neural Networks for Nonlinear Self-Tuning Adaptive Control",** IEEE Control Systems Magazine, pp. 44-47, April 1990.