1997 IEEE International Symposium on Circuits and Systems, June 9-12, 1997, Hong Kong

High Speed On-line Neural Network Control of an Induction Motor Immune to Analog Circuit Nonidealities

Jin Liu, *B. Burton, F. Kamran, M. A. Brooke, *R. G Harley, T. G. Habetler
Georgia Institute of Technology, USA, *University of Natal, South Africa

Abstract - **A neural network using the Random Weight Change algorithm is shown able to be trained to perform on-line control of the current of an induction motor stator, despite analog circuit nonidealities. The induction motor is a complex nonlinear electromechanical system, with rapidly time-varying system parameters. Due to the small time constant of this power electronic system, the neural network must be able to finish each training cycle in less that 50 microseconds, which is only possible when controlled by specifically designed hardware circuits. An analog circuit is preferred for its ability to implement a reasonable size of network on one integrated chip. The analog circuit nonidealities are overcome by the Random Weight Change (RWC) algorithm. RWC is based on the method of random searching, and achieves similar performance to the back-propagation (BP) algorithm. The back-propagation algorithm is very difficult to implemented in analog hardware due to its sensitivity to offset and nonlinearity errors, the RWC algorithm is simulated with analog circuit nonidealities, and is shown immune to these problems, thus the RWC algorithm is found ideally suited for the high speed analog circuit neural network implementation.**

I. INTRODUCTION

In this paper, we present research in controlling nonlinear, rapidly time-varying physical systems, using analog hardware neural networks.

Neural networks have been successfully applied in many control areas, such as controlling robot arms, chemical process control, continuous production of high-quality parts, and aerospace applications [1]. The application we present in this paper is the nonlinear adaptive control of induction motor drives, which will be explained in more detail in section IV. In order to control such systems, the control action must be taken at least every 100 μsec. The requirement of high speed nonlinear adaptation is typical in many other power electronic applications, as well as in the area of engine control. This means that these control applications are difficult or impossible to implement by current methods. Excellent work by [2] [3] [4] etc., show that neural networks can be used to model and control complex nonlinear physical systems with unknown or slowly varying plant parameters. This is because neural networks have the ability to learn complex input-output mappings without a detailed analytical model of the system. However, the application of our interest has an extra feature, that is rapidly time-varying plant parameters. It thus requires a very fast adaptive neural network system to control it on-line.

To achieve fast adaptation on-line, a hardware implementation is preferred. Because control done by a software program on a computer is usually not fast enough to keep up with the systems considered here, and thus cannot offer meaningful control. There are certainly some advantages of software control, like precise implementation of the algorithms. However it is not always convenient to install computers (fast enough computers are very expensive, and have a large volume) everywhere, or to carry them around an aircraft. So analog hardware circuit implementation of the neural network is proposed to offer fast speed and compact size.

Again, there are digital circuits and analog circuits. Digital circuits have the advantage of implementing precise computation via adders and shifters, which are basically the same as a computer, running software control. While analog circuits [5] [6] [7] [8] experience the nonidealities, like one-side non-linear multipliers, leakage of the weight storage, etc. On the other hand, a neural network of sufficient size can be implemented on a single integrated chip using an analog circuit, [11] while to implement the same size neural network, it requires a lot more area to achieve the desired bits of precision in a digital implementation. As we know, the bigger the chip size, the lower the yield factor, the more expensive the product will be.

An Analog circuit is preferred because using the Random Weight Change algorithm[9], the learning is insensitive to the analog circuit nonidealities, while achieving similar performance to back-propagation. Back-propagation is almost impossible to be implemented with an analog circuit, because it requires precise computations of derivatives, and multiplication. Also, the RWC algorithm tends to escape from the trap of local minimums by changing the weights in different random directions. Trapping in a local minimum is one drawback of the back-propagation algorithm.

The Random Weight Change Algorithm will be presented in section II, with some simulation results. In section III, the induction motor drive system is presented, software simulation of neural network control without any analog circuit nonidealities are show. These simulations use a modified version of the RWC algorithm. In section IV, analog circuit nonidealities are added to the neural network controller of the induction motor drive. The simulation result shows that the RWC neural network is insensitive to these analog circuit nonidealities. In the last section, we present the conclusion, and propose future work.

II. RANDOM WEIGHT CHANGE ALGORITHM

In this section, we describe the Random Weight Change algorithm and show that it is insensitive to nonlinearity in the neural network weights. This algorithm is a learning algorithm for multilayer neural networks similar to the well known back-propagation algorithm. Unlike the back-propagation algorithm, which makes the error function of the network decrease in the direction of the steepest descent, the RWC algorithm makes sure that the error function decreases on average, but it may go up or down at any one time. So the minimum point is reached without tracing the steepest slope of the error function. The learning defined by RWC is as follows:

$w_i\,(n+1) = w_i\,(n) + \Delta w_i\; i(n+1);$

if error decrease,

$\Delta w_i\,(n+1) = \Delta w_i\,(n);$

if error increase,

$\Delta w_i = \text{rand}(n)\,,$

where rand(n)= $+\,\delta$ or $-\,\delta$ with equal opportunities.

Three applications: XOR gate; 3 bit A/D conversion; and Target localization were simulated in software, using a nonlinear multiplier. The multiplier is characterized by $y = x * (1-x/3)*w$, where y is the product, w is the weight, and x is the input.

The result of these simulations with BP and the RWC algorithm are reproduced on Table 1. It shows that RWC learns slower than BP, for the first two applications. For the third application, which is more complicated, BP fails to learn, while RWC still learns. From these results, we can see that precisely computed derivatives, and ideal multipliers are necessary for BP but not for the RWC algorithm.

Table 1. Simulation of BP and RWC with nonlinear multipliers

Problems			BP	RWC
XOR Gate			1240	3100
3 bit A/D conversion			117400	325000
Target Localization	Analog	r	no learning	397000
		θ	no learning	449000
	8 bits Digital	θ	no learning	492000

Also, due to the randomly changed weights, this algorithm has the advantage over back-propagation, in that it tends to escape from a local minimum if trapped in them.

We have given an overview of the RWC algorithm in this section, in next section, RWC is applied to learning the control of induction motor system.

III. INDUCTION MOTOR SYSTEM AND CONTROL

The induction motor is a complex, multi-variable nonlinear dynamic electromechanical system. The objective of controlling an induction motor is to control the flow of power into the motor, so as to produce a torque on the rotor shaft, which will result in a desired shaft speed. In order to produce the desired torque, the stator current is first calculated. Based on the required stator current, the applied voltage is then calculated. The relationship between the applied voltage, resultant current, torque and speed is as shown in Figure 1.

Voltage → Current → Torque → Speed

Figure 1. Relationship between the stator voltage, current, shaft torque and speed. The applied current is the input, and the generated speed is the output.

The relationship between the applied stator voltage and the resultant stator current vector is nonlinear, with respect to the speed of the motor shaft. This relationship is also implicitly time varying due to electrical parameter variations. For example, the inductance can vary significantly and rapidly due to significant and rapid changes in current magnitudes, the AC resistance of conductors is frequency dependent and can vary rapidly with rapid changes in current frequency, and the relatively slow changes in resistance due to heating effect.

In addition, the relationship between the stator current and the mechanical torque is also nonlinear and affected by time varying parameters. The shaft speed is determined not only by the net torque applied, but also the mechanical load .

Therefore, the dynamic performance of the motor depends on how well the controller can adapt to the parameter variations. The two layer neural network controller developed by Burton etc. [10] successfully controlled the induction motor to produce the desired current, using a modified version of the RWC algorithm.

The modified algorithm makes twenty random trials to change the weight in twenty different directions, and calculates the error for each trial. One trial is in the direction of the previous round. However, each change is not permanent, the permanent change is made in the direction of the trial with the least error. The underlining idea is the same as the algorithm explained in the previous section. The modified version improves the algorithm by making more trials, but takes a longer time for each trail.

The simulation of the neural network in [10] was based on ideal multipliers, and did not include any nonidealities of the analog circuit. One of the results of using the RWC algorithm in that work is shown in Figure 2. The horizontal axes is time in seconds, and the vertical axes is the current. In less than 0.1 second, the neural network learns to follow the desired current output. In this case, the simulation is carried out with ideal two sided multipliers. In the next section, we are going to show the performance of the neural network with non-ideal one-sided multipliers.

IV. ANALOG CIRCUIT NONIDEALITIES

Based on the fabricated circuit in [11], we now test the neural network controller for the induction motor with the one-sided non-linear multipliers. An ideal multiplier is expressed as

$$y = w * x * f(x).$$

It is two-sided, which means x can be positive or negative, and is linear, which means f(x)=1. However, in the condition of nonlinear, one-sided multipliers, as implemented buy the analog circuits in [11], x can only be positive, and f(x) is a nonlinear function of x. The ideal multiplier and the nonlinear, one-sided multiplier with offset are shown in Figure 3. The nonideal multiplier is defined by $y = w * x * (1-x^2/3)$.

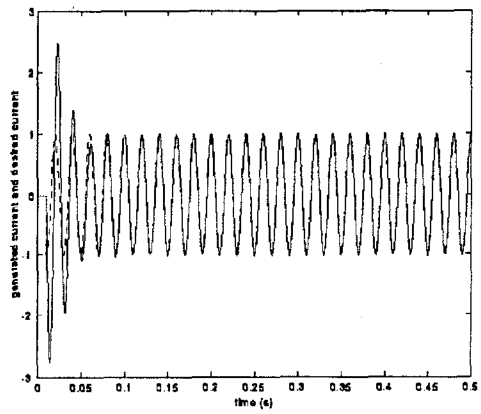

Figure 2. Simulation result of induction motor using RWC algorithm, the dotted line is the desired current, and the solid line is the actual current generated. Ideal multiplier is used in the simulation.

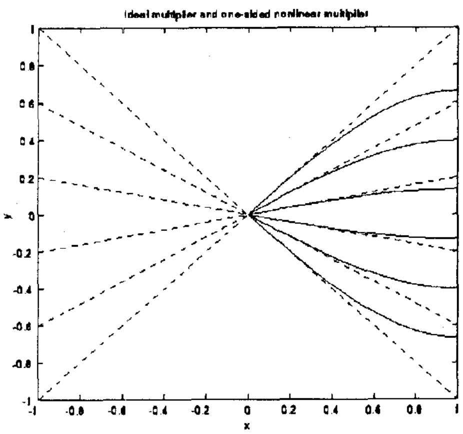

Figure 3. Ideal multiplier and the one-sided nonlinear multiplier. The dotted line characterizes the ideal two-sided multiplier, and the solid line is the nonideal multiplier typical for analog circuit implementation in [11].

Substituting the ideal multiplier with the non-ideal multiplier in Figure 3, we simulated the performance of the neural network controller stated in the previous section. The result of the generated current is shown as the solid curve in Figure 4, the dotted curve is the generated current using an ideal multiplier. It is the same current as shown in Figure 3, and is redrawn here for comparison.

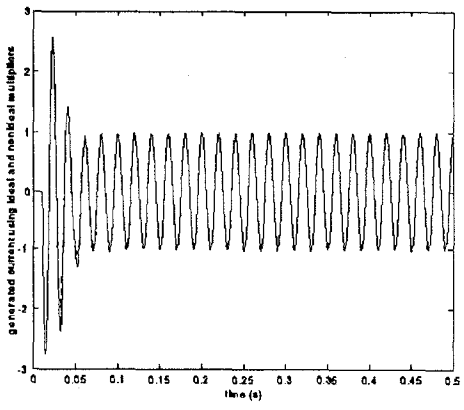

Figure 4. Comparison of generated current using ideal multipliers and the nonideal multipliers typical for analog circuit implementation.

As we can see, the current generated using the non-ideal multiplier is almost the same as that using the ideal multipliers. It shows that the neural controller using the RWC learning algorithm is immune to the nonideality of the multipliers, which was the major problem of BP learning when applied to learning neural networks built from analog circuits.

V. CONCLUSION AND FUTURE WORK

The Random Weight Change algorithm is suitable for analog hardware implementations of neural networks that can control rapid time-varying physical systems, despite the analog circuit nonidealities.

Future work now involves that following steps. (1) Test a prototype analog neural network chip with a software simulation of an induction motor system, make appropriate adjustments on the chip according to the test result. (2) Test the chip with a real physical system and make appropriate adjustments.

REFERENCE:

[1] Werbos, P.J., "Neurocontrol and elastic fuzzy logic: capabilities, concepts, and applications," IEEE Transactions on Industrial Electronics, vol. 40, no. 2, pp. 170-80.

[2] Ahmed, R.S., Rattan, K.S. and Khalifa, I.H., "Real-time tracking control of a DC motor using a neural network," Proceedings of the IEEE 1995 National Aerospace and Electronics Conference, vol. 2, pp. 593-600.

[3] Shaffner, and C., Schroder, D., "An application of general regression neural network to nonlinear adaptive control," Fifth European Conference on Power Electronics and Applications, vol. 4, pp. 219-24.

[4] Kalanovic, V.D. and Tseng, W.-H., "Back propagation in feedback error learning," Proceedings of Neural, Parallel and Scientific computations, vol. 1, pp. 239-42.

[5] Gillian Marshall and Steve Collins, "An analog radial basis function circuit incorporating floating-gate devices," Analog Integrated Circuits and Signal Processing, vol. 11, pp. 21-34.

[6] Michel Verleysen and Paul G. A. Jespers, "An analog implementation of the Hopfield's Neural Networks," IEEE Micro, December 1989, pp. 46-55.

[7] Carver Mead, "Analog VLSI and neural systems," Addison-Wesley Publishing Company, 1989.

[8] Olivier Rossetto, Christian Jutten, Jeanny Herault and Ingo Kreuzer, "Analog VLSI synaptic matrices as building blocks for neural networks," IEEE Micro, December 1989, pp. 56-63.

[9] Kenichi Hirotsu and Martin Brooke, "An analog neural network chip with random weight change learning algorithm," Proceeding of the IJCNN, October 1993, Japan, pp. 3031-3034.

[10] Burton, B., Kamran, F.; Harley, R.G.; Habetler, T.G., Brooke, M.; Poddar, R., "Identification and control of induction motor stator currents using fast on-line random training of a neural network," Conference Record of the 1995 IEEE Industry Applications Conference, vol. 2, pp. 1781-7.

[11] K. Hirotsu, M.Brooke. "An Analog Learning Neural Network IC," International Joint Conference on Neural Networks, Nagoya, Japan, October 29th 1993.