Dynamically Reconfigurable Neural Network Hardware Design for Ultrasonic Target Detection

Sungjoon Yoon, Erdal Oruklu, and J. Saniie
Electrical and Computer Engineering Department
Illinois Inst. of Technology
Chicago, Illinois 60616

Abstract - **Ultrasonic flaw detection and classification is an essential method for many NDE applications. Frequency diverse algorithms such as split spectrum processing (SSP) can be used effectively for flaw detection since they decompose the signal in different subbands and induce significant statistical variation in scattering noise or speckles. Following the subband decomposition, neural networks can be used as post-processors for enhanced detection performance. In this study, it has been shown that using neural networks (NN), the flaw-to-clutter ratio (FCR) improvement up to 13dB can be achieved when the input experimental signal has FCR equal to 0dB or less. The experimental ultrasonic flaw signals masked by grain scattering noise (i.e., clutter) are obtained from a steel block using a 5MHz transducer and 100MHz sampling rate. Neural network algorithms are computationally demanding and require substantial hardware resources for implementation due to their complex connectivity. Therefore, we have designed an efficient and adaptable neural network architecture for real-time realization of ultrasonic flaw detection applications. The hardware implementation takes full advantage of the inherent parallelism of neural networks and is dynamically reconfigurable; indicating the system is flexible and can be suitable for various NDE applications.**

Keywords - ultrasonic flaw detection; neural networks; Reconfigurable architecture

I. INTRODUCTION

Ultrasonic target detection and classification is a difficult problem due to high microstructural scattering noise. Neural networks [1] with split spectrum processing (SSP) are a suitable solution for improved ultrasonic imaging. Hardware implementation for real-time ultrasonic application is another major problem. In this paper, we present a reconfigurable neural network (NN) architecture for ultrasonic target detection. Ultrasonic signals can have different flaw-to-clutter ratios (FCR) due to various environmental and measuring conditions. The parameters of NN need to be adjustable for different conditions of ultrasonic measurements. Therefore, the architecture of the neural network must be designed to allow dynamic reconfiguration for various numbers of hidden nodes and weight coefficients.

In the application of ultrasonic target detection, SSP is an effective method. Split spectrum processing (SSP) divides the signal frequency band into several smaller frequency bands. The ultrasonic grain scattering signal is randomly distributed across the frequency bands. In contrast, the echo backscattered from the flaw has a persistent presence in the majority of the frequency bands. Therefore, the output from the SSP algorithm can be used by a neural network system to differentiate the grain echoes from the flaw echoes in order to enhance the FCR.

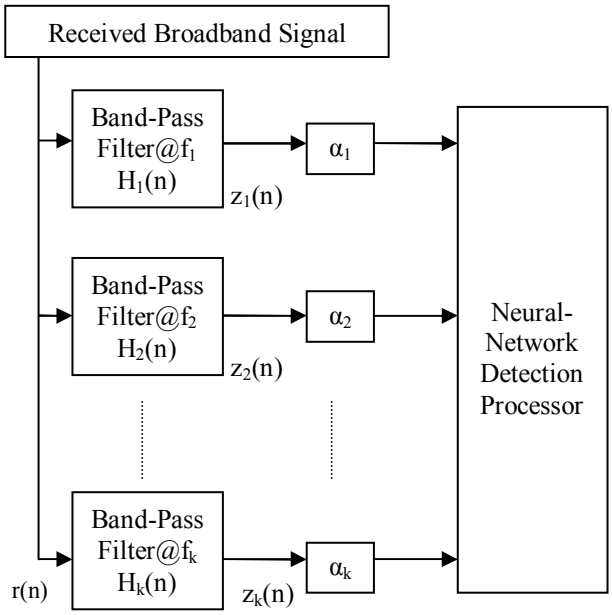

Figure 1. Block diagram of Split-Spectrum Processor followed by the Neural Networks for ultrasonic target detection

The block diagram of an ultrasonic target detection system using SSP is shown in Figure 1. The ultrasonic signal, r(n), is an input to SSP which is composed of bandpass filters. The output of the bandpass filters are scaled by α_i . The scaling factor, α_i , is used to normalize the SSP output to obtain the equally powered output signal on the SSP channels. The outputs of bandpass filters are passed into a Neural Network detection processor. Neural networks are widely used for signal classification due to their trainability and adaptability. In ultrasonic target detection, neural networks can detect flaw signal after training. A diagram of a three-layer feedforward

neural network is shown in Figure 2. The three-layer feedforward neural network which can perform a complex nonlinear mapping process [2] is used as a detection processor. The neural nodes of the hidden layer receive the weighted inputs from the input layer and then perform the activation function. The neural nodes of the output layer sum up the output of the hidden layer.

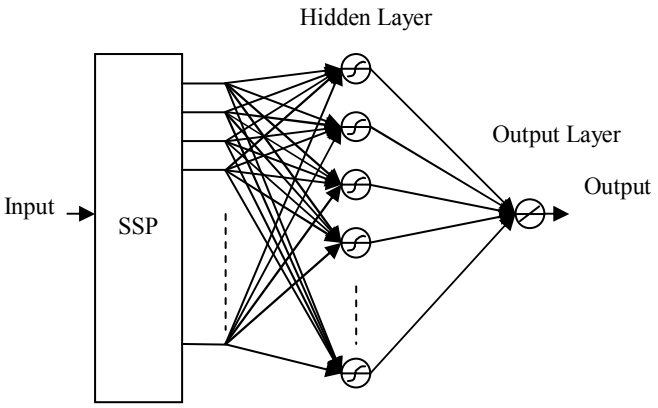

Figure 2. A three-layer feedforward neural network with SSP

The general neural node model can be expressed by

$$y_j = \varphi(\sum_i w_{ji} x_i + b_j) \qquad (1)$$

where x_i is a set of inputs of each neuron, y_j is a set of outputs of each neuron, and b_j is a set of bias of each neuron. Each input is multiplied by a weight coefficient w_{ji}. The subscript *ji* refers to the input *i* in neuron *j*. The term φ is an activation function. The activation function used in the hidden layer is the sigmoid function.

$$\varphi(x) = (1 + e^{-x})^{-1} \qquad (2)$$

The learning process is needed to classify ultrasonic flaw signal when using neural networks. The learning process allows neural networks to learn the environments of particular applications. The learning step takes place through iterative process of adjusting weight values. In this paper, an off-chip learning (i.e., not implemented in hardware) is applied in order to obtain weight and bias coefficients.

An experimental result using only software to implement the NN is shown in Figure 3. The experimental data was acquired using a broadband transducer with a 5 MHz center frequency. 10 channel Gaussian filters are used for SSP. The NN target detector has 5 hidden nodes and 1 output. The FCR of the original data is zero dB. The NN target detector improves the visibility of the flaw echo significantly resulting in a FCR of 23dB.

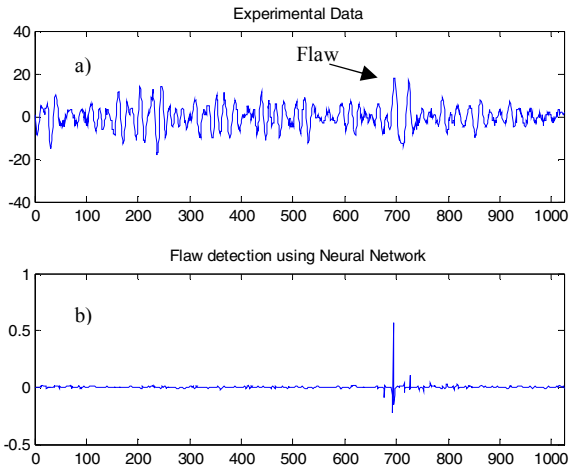

Figure 3. Flaw detection using NN
a) Ultrasonic experiment measurement
b) NN processing result using software realization

II. Reconfigurable Neural Network Architecture

The architecture of an ultrasonic target detection processor using NN is shown in Figure 4. The ultrasonic data is stored in the Input Memory. This data is fed to the SSP Block which can be implemented by Fast Fourier Transform (FFT) or other efficient filter design [3]. The outputs of SSP are stored in the Intermediate Memory. The Neural Network Block performs the detection process. The Embedded Microcontroller configures and assigns weight and bias coefficients to neural network block during initialization.

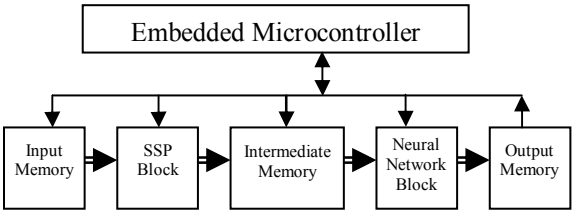

Figure 4. The diagram of ultrasonic target detection processor

The architecture of the neural network is composed of processing element (PE) arrays. A PE architecture is shown in Figure 5. Each PE has a Multiplier, an Adder, a Register, a RAM block and two Multiplexers. The RAM block of PE stores weight coefficients. A register is used for the pipeline of the multiplier. The pipeline register is enabled or disabled by Mux1. Mux2 allows to choose an adder if necessary.

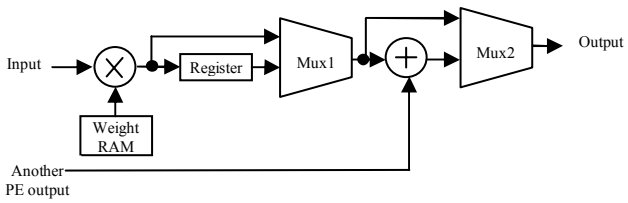

Figure 5. PE Architecture

The PE arrays shown in Figure 6 are used to assemble the hidden layer as well as the output layer for the neural networks. The hidden layer in Figure 6 is made of PEs, a Multiplexer, a Bias RAM, Adder, Sigmoid Function logic and a Register; whereas the output layer utilizes a single PE, a Multiplexer, a Bias RAM, Adder and a Register. PEs are serially connected in the hidden layer. The Sigmoid Function logic is implemented by piecewise linear approximation [4] in order to simplify the hardware requirements. The adder is required for the serial computation mode when the number of PEs is not enough (i.e., less than the number of inputs times the number of hidden layer nodes) to compute the output of neural networks within a clock cycle. It is possible to operate fully parallel to maximize the computation speed. However, this would require a large number of PEs. The computation mode (parallel or serial) and the PE array configuration can be dynamically reconfigured during the system initialization. The number of bits for input and weight coefficients is 8 bits. 16 bits are used for computational results. In order to minimize hardware resources with reasonable detection performance, the number of bits for input data, weight coefficients, and computational results was evaluated using MATLAB simulation.

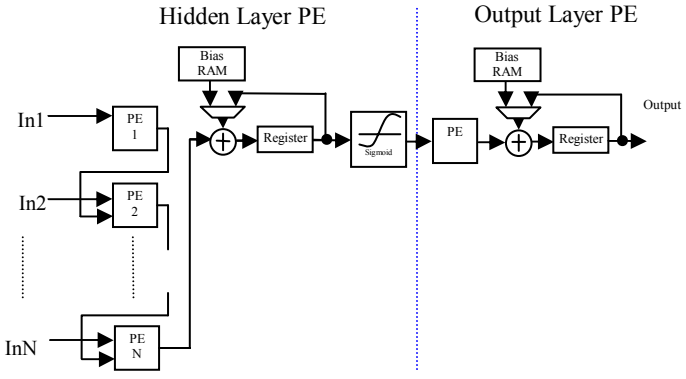

Figure 6. Neural networks implementation with PE arrays

A nonlinear sigmoid function is hard to implement in hardware. Therefore, an efficient piecewise linear approximation of a nonlinear (PLAN) function is used [4] to realize the hardware for the sigmoid function. In the PLAN method, the sigmoid function is approximately made of various straight lines (for clarification see Figure 7 and Table I). Complexity of computing the nonlinear approximations is reduced by choosing shifter instead of multiplier. The shift and add operation are implemented with minimal logic gates.

In this study, a neural network block is implemented with 5 inputs and 5 hidden PEs. This hardware is used to compute neural networks with 10 inputs, 5 hidden nodes and 1 output for ultrasonic target detection.

TABLE I. IMPLEMENTATION OF PLAN
X : INPUT OF SIGMOID FUNCTION,
Y: OUTPUT OF SIGMOID FUNCTION

Condition Input X	Operation Output Y
|X|>=5	Y=1
2.375=<|X|<5	Y=0.03125*|X|+0.84375
1=<|X|<2.375	Y=0.125*|X|+0.625
0=<|X|<1	Y=0.25*|X|+0.5
X<0	Y=1-Y

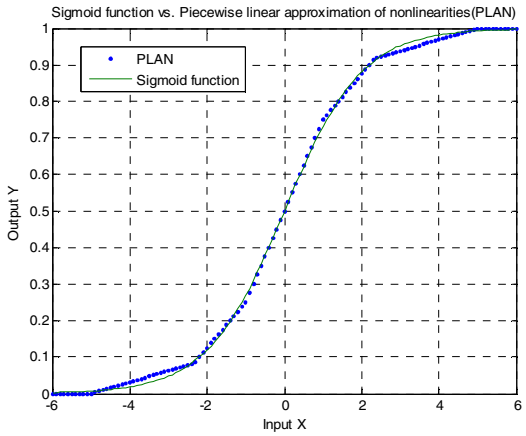

Figure 7. Sigmoid function vs. piecewise linear approximation

Each hidden node needs 10 multiplications due to 10 inputs. Therefore, 2 clock cycles are needed to compute serially 1 hidden node since the hardware has 5 PEs and the number of inputs is 10. This serial computation is implemented by input multiplexing and accumulation in the hidden layer. Hence, the total computation time for 10 inputs and 5 nodes in the hidden layer requires 10 clock cycles.

III. HARDWARE IMPLEMENTATION RESULT

The neural network block is implemented in the FPGA of Virtex2Pro30 provided by Xilinx®. Verilog-HDL is used with an ISE Foundation program for NN design. The NN hardware

implementation has 5 inputs, 5 PEs in the hidden layers. The synthesis report shows that 738 slices, 128 flip flops and 6 multipliers are used and it can be operational at a maximum frequency of 56.294MHz.

The neural network hardware architecture was designed to have the same architecture used for the software realization of NN (5 hidden nodes, 1 output, 10 channels SSP, same weight and bias coefficients). The NN result of hardware implementation is shown in Figure 8. This hardware realization of the NN target detector improves FCR around 13dB and enhances the flaw visibility significantly. This result clearly demonstrates the feasibility of a hardware realization of a reconfigurable neural network for ultrasonic target detection. The FCR enhancement performance difference between software and hardware implementation is due to use of fixed-point numbers as opposed to full-precision used in software implementation.

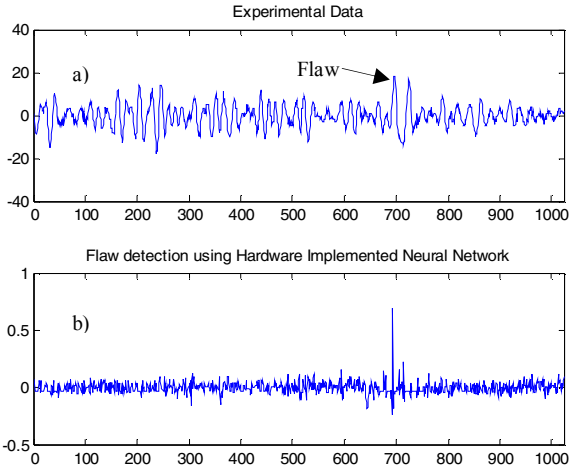

Figure 8. Flaw detection using a hardware NN
a) Ultrasonic experiment measurement
b) NN processing result using hardware realization

IV. CONCLUSION

In this investigation, we present reconfigurable hardware architecture of neural networks for ultrasonic imaging and target detection applications. The neural network parameters such as weight coefficients and the number of neural nodes are dynamically reconfigurable without any hardware modification for applications in different testing conditions. Furthermore, this architecture demonstrates the effectiveness of neural networks for ultrasonic imaging.

REFERENCES

[1] H.C. Sun and J. Saniie "Nonlinear signal processing for ultrasonic target detection" , IEEE Ultrasonics Symposium Proceedings, pp. 855-858, 1998.

[2] Richard P. Lippmann, "An introduction to computing with neural nets", IEEE ASSP Magazine, pp. 4-22, April 1987.

[3] Erdal Oruklu, Fernando Martinez Vallina and Jafar Saniie, " Efficient hardware realization of frequency-diverse ultrasonic flaw detection using zero-phase IIR filters" , IEEE Ultrasonics Symposium Proceedings, pp 1785-1788, 2005.

[4] H.Amin, K.M.Curtis, and B.R.Hayes-Gill, "Piecewise linear approximation applied to nonlinear function of a neural network", IEEE Proc-Circuits Devices Syst., Vol. 144, No. 6, pp. 313-317, December 1997.

[5] Daniel Ferrer, Ramiro Gonzalez and Roberto Fleitas, Julio Perez Acle, Rafael Canetti, "NeuroFPGA – Implementing Artificial Neural Networks on Programmable Logic Devices", IEEE Proceeding of the Design, 2004.

[6] Ma Xiaobin, Jin Lianwen, and Shen Dongsheng and Yin Junxun, "A mixed parallel neural networks computing unit implemented in FPGA", IEEE Int. Conf. Neural Networks & Signal Processing, pp. 324-327, December 2003.