A Parallel Hardware Multiple Microprocessor Asynchronous Neural Network Architecture for Control Applications

James Shippen, Ronald Westra* and Nicholas Freear
The School of Manufacturing and Mechanical Engineering
University of Birmingham, Birmingham, B15 2TT
The United Kingdom
Phone: +44-121-414-4153, Fax: +44-121-414-3688
email: {j.m.shippen, n.d.freear}@bham.ac.uk
*The Department of Mathematics
Maastricht University
The Netherlands
Phone: +31-43-388-3722, Fax: +31-43-321-1889
email: westra@math.unimaas.nl

ABSTRACT: This paper presents an architecture for a neural network which has been constructed from multiple asynchronous microprocessors. The architecture enables indefinite extension of the network to model transfer functions of ever increasing complexity however the parallel nature of the hardware results in the complexity having a minimal affect on processing speed. The network is capable of learning a transfer function which, by the use of a transport delay, can represent temporal characteristics of a dynamical system and hence form the kernel of an motion controller. The network was constructed from 50 parallel asynchronous PIC16C57 microprocessors using the in-star and out-star learning paradigms. A motion controller based on an anticipative strategy was constructed which required no explicit mathematical model of the controlled system, is adaptive and is readily applicable to highly non-linear systems.

KEYWORDS: Neural network, control, parallel architecture, printing

INTRODUCTION

High speed industrial systems typically require sophisticated controllers to ensure optimum performance of the process and its stability in the presence of unknown disturbances. One example is the pad printing process where the quality of the printed graphic is dependant on a large number of variables, for example ambient temperature, humidity, static charge, contact load, printing rate, ink viscosity. Following automated visual inspection of the printed graphic, many of these variables can be manipulated to maintain or improve the quality of process.

This paper presents results of research into a suitable controller for the pad printing process. The input to the controller is the output from a diagnosis/prognosis system which demands motion from various actuators incorporated into the pad printing machine.

CONTROL STRATEGY

The pad printing process requires the control of a number of actuators. The literature contains many different control strategies however this problem required an approach which could accommodate multiple input, adaptive to machine-to-machine variation and ageing and capable of operating within a non-linear environment. It was therefore decided to utilise an anticipative neural network controller.

The anticipative controller maintains a model of the dynamic characteristics of the system which it is controlling. This model can then be used to predict the outcome of various mutually exclusive control demands. Given a penalty function, the most appropriate action can then be selected and implemented. This process this then repeatedly executed faster than real time to predict further into the future. The mathematical model of the system could be based on a detailed knowledge of the system resulting in a totally analytic model (for example in terms of Laplace transforms) but this will only be applicable to simplistic systems with little non-linearity nor variations between examples of nominally identical systems. The current application requires a more robust approach in the absence of a well defined *a-priori* mathematical model. Therefore it was decided to adopt a mathematical model which did not required an explicit statement of the system's characteristics.

An open-loop actuator was driven by a random demand signal which spanned the expected demand signals of the operational actuator. This resulted in a twitching motion of the actuator. The position and velocity of the actuator was monitored at regular intervals. The collection of current position, current velocity, demand signal, position one interval later and velocity one interval later formed the training set for a software neural network The software neural network ran on a conventional serial controller. This neural network could then be used to predict the motion of the actuator in the future.

However for the anticipative control strategy to be effective it is required that the mathematical model can predict motion faster than real time. It was therefore decided to implement a hardware parallel neural network.

PARALLEL HARDWARE NEURAL NETWORK ARCHITECTURE

All nodes in a software neural network have implicit access to global knowledge of all of the weights in the network. For the network to compete with conventional serial controllers a network containing a large number of nodes would be required. This would be facilitated by an architecture which could have learning capability based on only local knowledge of the networks weights and hence a modular approach could be adopted enabling infinite network expansion.

A suitable architecture is based on the in-star and out-star learning algorithms as shown in figure 1. Each node consists of a Parallax Basic2 stamp consisting of a PIC16C57 micro-controller, PBASIC2 interpreter, 2048byte EEPROM and 32bytes of RAM. The learning and operating code for the in-stars and out-star was stored on the each node's EEPROM as appropriate and the weights were stored in its RAM. The input and target outputs were digitised by an Analogue Devices AD7575 and the output was returned to an analogue signal by an Analogue Devices AD9547. It was required that communication between the nodes be achieved by a single wire and therefore numbers were transmitted around the network as pulses of varying length with a 2μs resolution. There is no central controller of this network and all of the nodes run asynchronously.

The in-star learning rule resulted in the in-star nodes weights adapting to recognises input patterns – these nodes are self organising and do not require a target output. The out-star learning rule modified the weights of the nodes to provide the correct output for given the target output.

CONCLUSION

An anticipative controller using a software implementation of a feed-forward neural network has been constructed. This had been used to control the motion of an actuator. The actuator was of low-quality, high friction and backlash. The network learnt the actuator's dynamic characteristics from 'twitch' data. The hardware neural network has been constructed and trained by a conventional computer via a A-D/D-A board to reproduce an arbitrary transfer function. The next stage of the research will be to use the hardware neural network in the role of a motion controller.

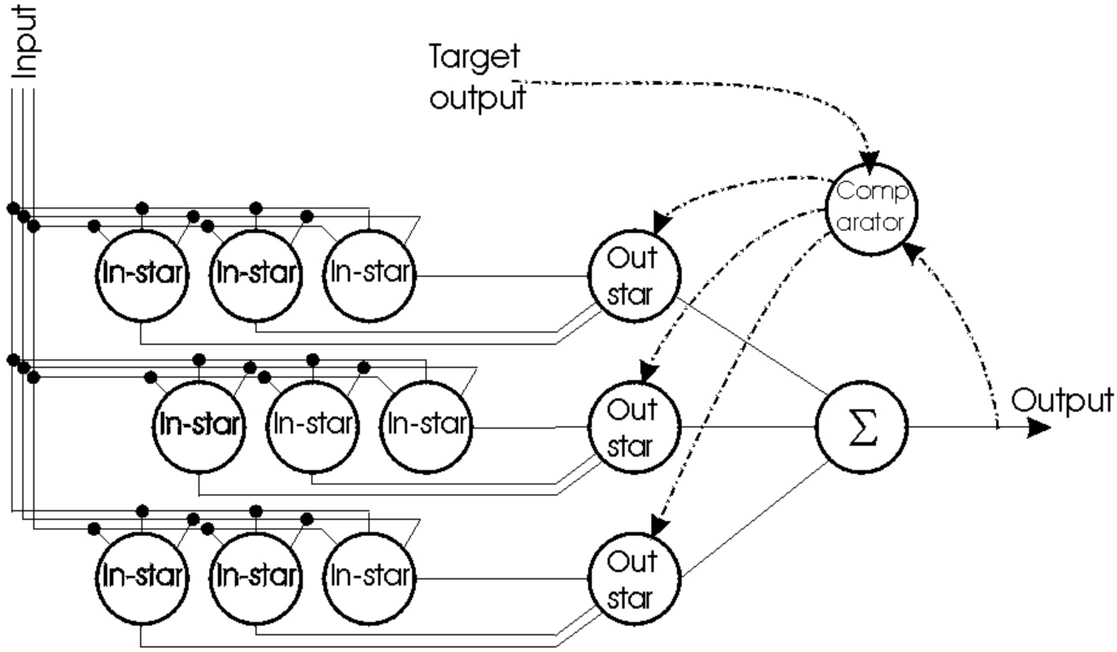

Figure 1: A hardware parallel multiple microprocessor asynchronous neural network architecture