Fermi National Accelerator Laboratory

FERMILAB-Conf-92/283

Drift Chamber Tracking with Neural Networks

Clark S. Lindsey, Bruce Denby and Herman Haggerty

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

October 1992

Submitted to *IEEE Transactions on Nuclear Science*, Special Issue for 1992 Nuclear Science Symposium

Disclaimer

DRIFT CHAMBER TRACKING WITH NEURAL NETWORKS

Clark S. Lindsey*, Bruce Denby, Herman Haggerty
Fermi National Accelerator Laboratory†
P.O. Box 500, Batavia, Il. 60510

Abstract

We discuss drift chamber tracking with a commercial analog VLSI neural network chip. Voltages proportional to the drift times in a 4-layer drift chamber were presented to the Intel ETANN chip. The network was trained to provide the intercept and slope of straight tracks traversing the chamber. The outputs were recorded and later compared off line to conventional track fits. Two types of network architectures were studied. Applications of neural network tracking to high energy physics detector triggers is discussed.

1 Introduction

With the very high event rates projected for experiments at the SSC and LHC, it is important to investigate new approaches to on line pattern recognition. The use of neural networks for pattern recognition in high energy physics detectors has been an area of very active research (see review in ref. 1). Charged particle tracking with neural networks, in particular, has been studied extensively. A major goal of these studies is to determine whether neural networks, which have highly parallel structures, could provide real time pattern recognition for triggering if they were implemented in hardware. At high energy collider experiments an interaction can produce a great number of tracks. In the central tracking devices, close to the interaction point, the large number of track and background signals make the track finding networks quite complicated, especially if there is track curvature due to a magnetic field. Recurrent type network architectures have been the most successful for track finding in large central detector events[2]. However, tracking with recurrent neural networks (all neurons interconnected and weights recalculated for each event) has only been carried out with off line data and with simulated networks. Hardware implementation of such networks, while not impossible, appears very difficult.

For some types of tracking chambers it is possible to apply simpler networks that are more amenable to hardware implementation. Drift chambers used in muon detection systems, for example, are typically constructed with only two to four layers of sense wires. The chambers, although covering large areas, lie far from the interaction region and so typically have lower occupancy rates than tracking chambers close to the interaction region. Since the chambers are usually outside the magnetic field and since they also only a few tens of centimeters thick, tracks crossing a muon chamber can be treated as straight lines. Using sense wire signals from a section of such a muon chamber, it is possible to use feed-forward neural networks with two layers of neurons to find track parameters. Such feed-forward neural networks are now available in VLSI. A fast muon trigger system could be built by combining the signals from many simple hardware networks, where each network is assigned to a section of the detector.

Here we present results of a study of a VLSI neural network for finding tracks in a muon drift chamber. A previous paper discussed tracking in a 3-layer muon chamber[3]. Here we give results for tracking in a 4-layer drift chamber. We show results for a couple of different neural network architectures and discuss possible implementation in a trigger.

2 Intel *ETANN* Chip

The Intel Electrically Trained Analog Neural Network (ETANN) has been described previously [4, 5]. The chip has 64 neurons or threshold amplifiers with sigmoidal response. Effectively, however, there are 128 neurons since the same 64 neurons are used for both the middle and output layers. A signal (analog voltage 0.0v to 3.5v) entering one of the 64 inputs is presented to a synapse. The output of the synapse is a differential current proportional to the multiplication of the input signal and a stored weight value. The current sum of the dot product of 64 inputs and 64 elements of a row of the input synapse array is presented to the neuron corresponding to that row. In addition there are 16 internal fixed (bias) voltages connected to each neuron. So each neuron sees a total sum of 80 voltage-synapse products. There are 80x64=5120 synapses in the first layer array plus 5120 synapses in the second layer array.

The synapse design is similar to a Gilbert multiplier cir-

*Current address: 1261 Foxan Ln., Aurora, IL 60506

†Fermilab is operated by the Universities Research Association under contract with the Department of Energy.

cuit. Here the difference in threshold voltages (Vt) of two floating gates provides the weight value:

$$I_{output} \approx (V_{input} - V_{ref_i}) \times (Vt_1 - Vt_2)$$

The weights values are limited to approximately ± 2.5 and have about 6 bit precision. Charge can be made to tunnel onto the floating gates with large voltage pulses. The synapses are non-volatile and can remain stable for several years. There are non-linearities in the voltage-weight multiplications, especially near the maximum weight and input voltage values[5]. However, these can be compensated for to some extent by the network training.

The neuron response, for a input reference voltage (V_{ref_i}) of 1.6v, is approximated by a sigmoidal function:

$$f_j(x_j) = \frac{3.0v}{1.0 + \exp(-x_j)} + 0.1v$$

where x_j is

$$x_j = G\left[\sum_k w_{jk}(V_k - 1.6v) + \sum_b w_{jb}(V_b - 1.6v)\right].$$

Here V_k is either the input voltage to the first layer or the output voltage of the middle layer neurons presented to the second layer. The w_{jk} is the weight for the connection between receiving unit j and sending unit k. Of the 16 internal biases ($V_b \simeq 4.0v$), seven are available for the user (the other 9 are reserved for the initialization of the chip by the development system described below.) The gain G can be varied with an external control voltage and was here set at roughly 1.0. A special binary mode with fast turn on from 0.0v to 5.0v is also available but is not suitable as a sigmoid for back-propagation training and was not used here.

After signals are presented at the inputs, the first layer neuron outputs will reach final levels within about $3\mu s$. The first layer outputs are then available on the output pins (e.g. for back-propagation calculations). Second layer processing is controlled by several external clock signals. First the neuron outputs are sampled and held and the inputs are disconnected from the first synapse array. Then the first level sampled outputs are presented to the second synapse array which in turn connects to the same neurons previously used for the first layer. The second layer processing takes up to $5\mu s$ for a total of about $8\mu s$ for 64 inputs, 64 neuron first layer, and 64 neuron second layer.

A pc-based development system for the ETANN is available[6]. The system allows one to do such things as initialise the chip, read weights or write weights to the chip, emulate the chip (e.g. with back-propagation trainer) and do chip-in-the-loop training (CIL). Normally one first trains with the emulator (here we used the *DynaMind* program[7]) and then, when the emulation performance is satisfactory, download the emulation weights to the chip. Some further CIL training is necessary since the emulation is not perfect. The synapses allow a limited number of weight changes before becoming degraded so doing the emulation reduces the number of weight changes required.

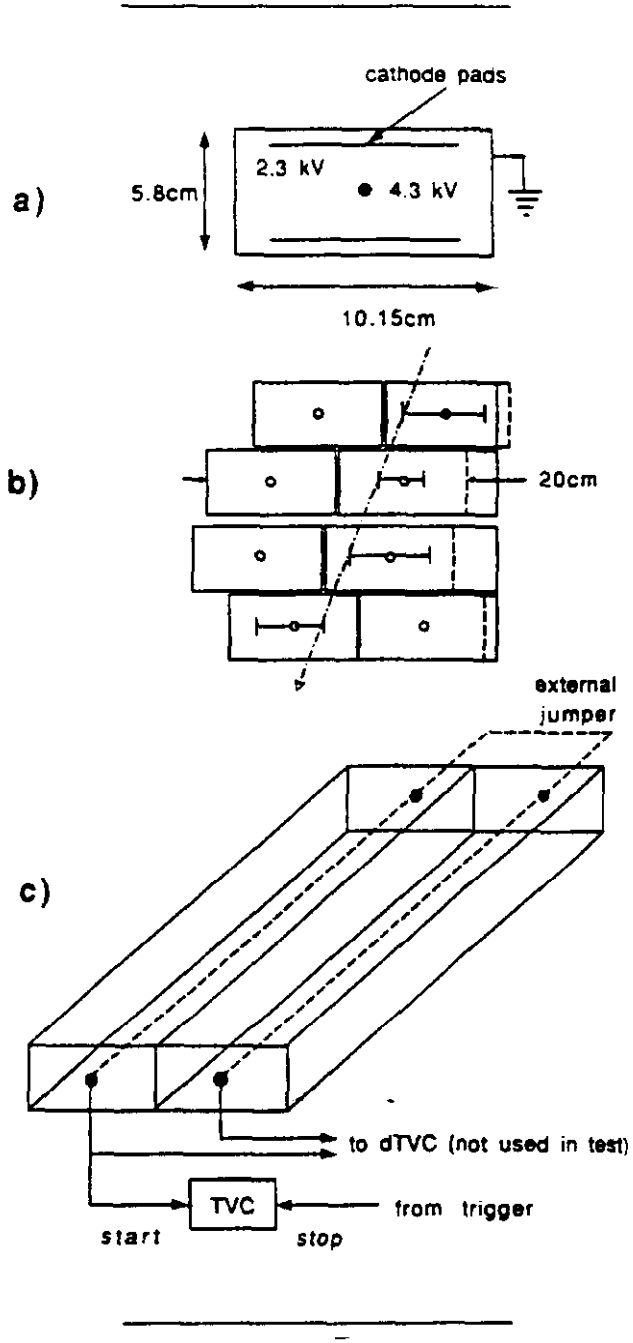

Figure 1: (a) Cross-section of a D0 muon chamber drift cell. (b) Track through a 4-layer chamber. Drift distances, with left-right ambiguities, are shown. The right edge cells are flush. Dashed lines show normal cell width. (c) Perspective view of a cell pair showing sense wire connection at one end, drift time and signal transit time electronics at other.

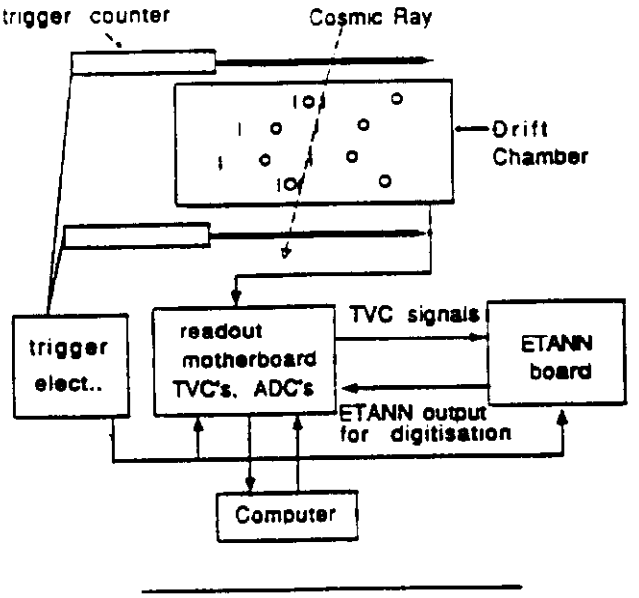

Figure 2: Test setup for tracking cosmic rays.

3 Drift Chamber Setup

Figure 1 shows diagrams of the muon chamber design used for the D0 collider detector system at the Tevatron [8, 9]. The cells have 5cm maximum drift in the sense wire plane, with maximum drift times of about $1\mu s$. The drift times are measured by time to voltage converters (TVC) and provide analog voltages proportional to the drift times (or distances). Above and below the sense wire are metal pads where voltages induced by the sense wire signal indicate where along the wire the track passed. As shown in figure 1c the sense wires are connected in pairs. Differences in signal arrival times at each end also indicate the longitudinal position of the wire. A combination of the pad and dTVC information gives an improved resolution. Here we only used the pad signals to set latches to indicate which of the two cells the track traverse. The prototype chamber used is only about a meter long so the transit time is negligible compared to the drift time.

Proceeding outward from the beam line, the D0 central muon trigger system consists of first a 4-layer drift chamber just outside of the central tracking chambers and calorimeters. Then there is a layer of magnetised iron and finally two 3-layer chambers. Here we examined tracking in a prototype 4-layer muon chamber which had 8 cells arranged in the pattern found on the edge of the full scale chambers (which can have up to 96 cells). Figure 2 shows the setup used to measure cosmic ray tracks. Two scintillators provide the trigger for an event and generate the stop signal for the TVC's. The TVC values vary linearly from 3.2v for tracks at the wire to 0.0v at the cell wall. The TVC outputs are picked off the analog bus on the muon chamber readout board and presented to the ETANN inputs. The chip outputs were then placed back on the bus and digitised along with the input voltages. The ADC values for the drift times and chip outputs and the pad latch values are all read out by the data acquisition system. Least square fit parameters were calculated off line (the chamber resolution is about $500\mu m$) from the drift chamber data and compared to the outputs of the ETANN.

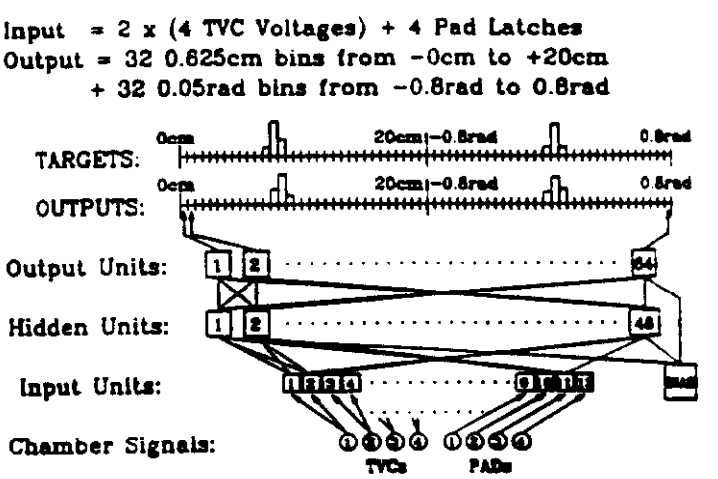

Figure 3: Two layer feedforward neural network with the *distributed* output technique for giving slope and intercept of tracks from drift chamber signals.

4 Neural Nets for Drift Chambers

Figure 3 shows one neural network architecture to determine the intercept and slope of muon chamber tracks. TVC signals from the four cell pairs shown in figure 1 are presented to the chip twice. As mentioned above, the maximum weight is ± 2.5 for the ETANN. Repeating the inputs effectively increases the maximum weight to ± 5.0 and improves the performance of the net. Also, pad latches from four cells are also input to the net. Here, for a given cell pair, a pad latch value of 3.0v indicates the hit is in the right hand cell, 0.1v indicates the hit is in the left hand cell as seen in figure 1.

The middle layer of the net in figure 3 has 48 neurons and the output layer has 64 neurons. The intercept and the slope are expressed as *bumps* in the distribution of output neuron activations. The first 32 neurons are used for the intercept and the second set of 32 for the slope. Each

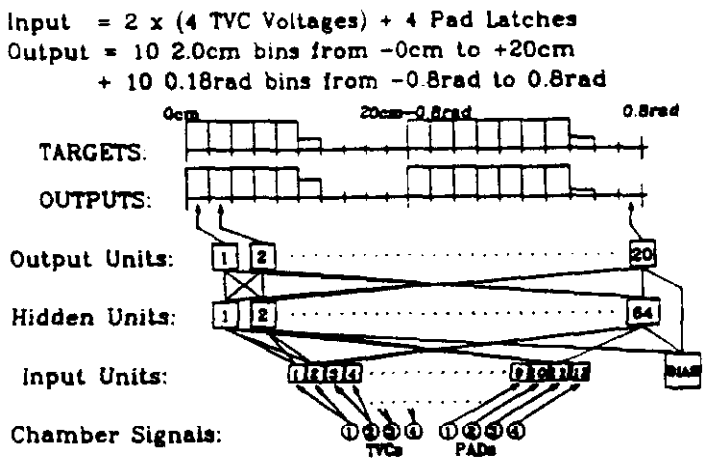

Figure 4: Two layer feedforward neural network with the *proportional* output technique for giving slope and intercept of tracks from drift chamber signals.

intercept neuron acts as a 0.625cm *bin* for the intercept range between 0.0cm and 20cm for the crossing point in the sense wire plane of the second layer from the top (figure 1). The slope neurons represent 50mrad bins between -0.8rad and +0.8rad, where the slope is give as the angle from vertical. The track intercept and slope are then given as the center of Gaussian bumps across 3-4 neurons. In practice the maximum output is found and an average over ± 2 neurons around the maximum is calculated.

The advantage of the *distributed* output method shown in figure 3 is that the averaging over the outputs makes the values *less sensitive to the jitter in single neurons*. Also, as discussed later, more than one bump can indicate more than one possible answer when there are ambiguities. If combined with other information, say, from another detector, having additional information can be useful in some applications[10]. The disadvantages of such a net include the large number of output neurons needed and the need for a 2nd circuit to calculate the averages.

One could have a two outputs architecture where one neuron has an activation proportional to the intercept and the other neuron has activation *proportional* to the slope. However, jitter, due both to electronic variability and to imperfections in the training of the network, limits the precision of such a net. To take advantage of the simplicity of proportional outputs but to limit the effect of neuron jitter, we tried the architecture shown in figure 4 (inspired by ref. 11). Here there are 10 outputs for the intercept and 10 for the slope. The sum of the first 10 outputs is proportion to the intercept and the sum of the second 10 outputs is proportional to the slope. The final sum is then less dependent on small variations in the individual outputs (most activations are either driven all the way on or all the way off.) A summing amplifier could provide a final single proportional voltage.

For each network type, files of 20000 track patterns were generated by a Monte Carlo program that sent tracks across the 8 cells at random angles and intercepts (but requiring at least one cell hit in each layer.) A given pattern consisted of both the simulated TVC and pad latch values and the target outputs. A back-propagation program on a workstation was run on these files for several million iterations. The resulting weight files were then transferred to the PC emulation program, which ran more slowly than the workstation program but did a more accurate simulation of the chip. The emulation did several tens of thousands of back-propagation iterations. Finally, the emulation weights were downloaded to the chip and a few thousand CIL iterations were carried out.

5 Results

Figure 5 shows four cosmic ray tracks in the D0 prototype chamber. The fit track and the neural network track are compared. The network was the distributed output net of figure 3. The activations of the 32 intercept neurons and 32 slope neurons are also shown, along with the fit values indicted by + symbols. In figures 5a-c the bump positions match well with the fit parameters. In figure 5d there was an ambiguity and the net gave a smaller output at the fit value than at the ambiguous track parameter. For the 3-layer network discussed in ref. 3 these ambiguous cases occur in 5-10% of the events. The extra layer here reduces the ambiguous cases to less than 0.5% of the cosmic ray tracks.

Figures 6a-b show distributions of the intercept and slope values from the fit and from the chip. The requirement of hits in all four layers causes the variations from uniform flat distributions. Figures 6c-d show distributions of the differences between fit track and NN track parameters for the distributed network architecture. The sigmas of the Gaussian fits give resolutions of 0.76mm for the intercept and 12mrad for the slope when compare to fit tracks with chi-square of less than one. For all tracks the resolutions are 0.99cm and 14mrad, respectively.

The proportional network show in figure 4 was also implemented in an ETANN chip. The 10 outputs for the intercept and the 10 outputs for the slope were added off line rather than added by a summing amplifier. Figures 7a-b show examples of cosmic ray tracks found by the off line fit and compared to the network output. The sum of the intercept and slope outputs are illustrated below each event pattern. The network values are represented by the length of horisontal bars and are compared to the fit values. Figure 7b shows the response to an ambiguous case. The network split the difference between the two

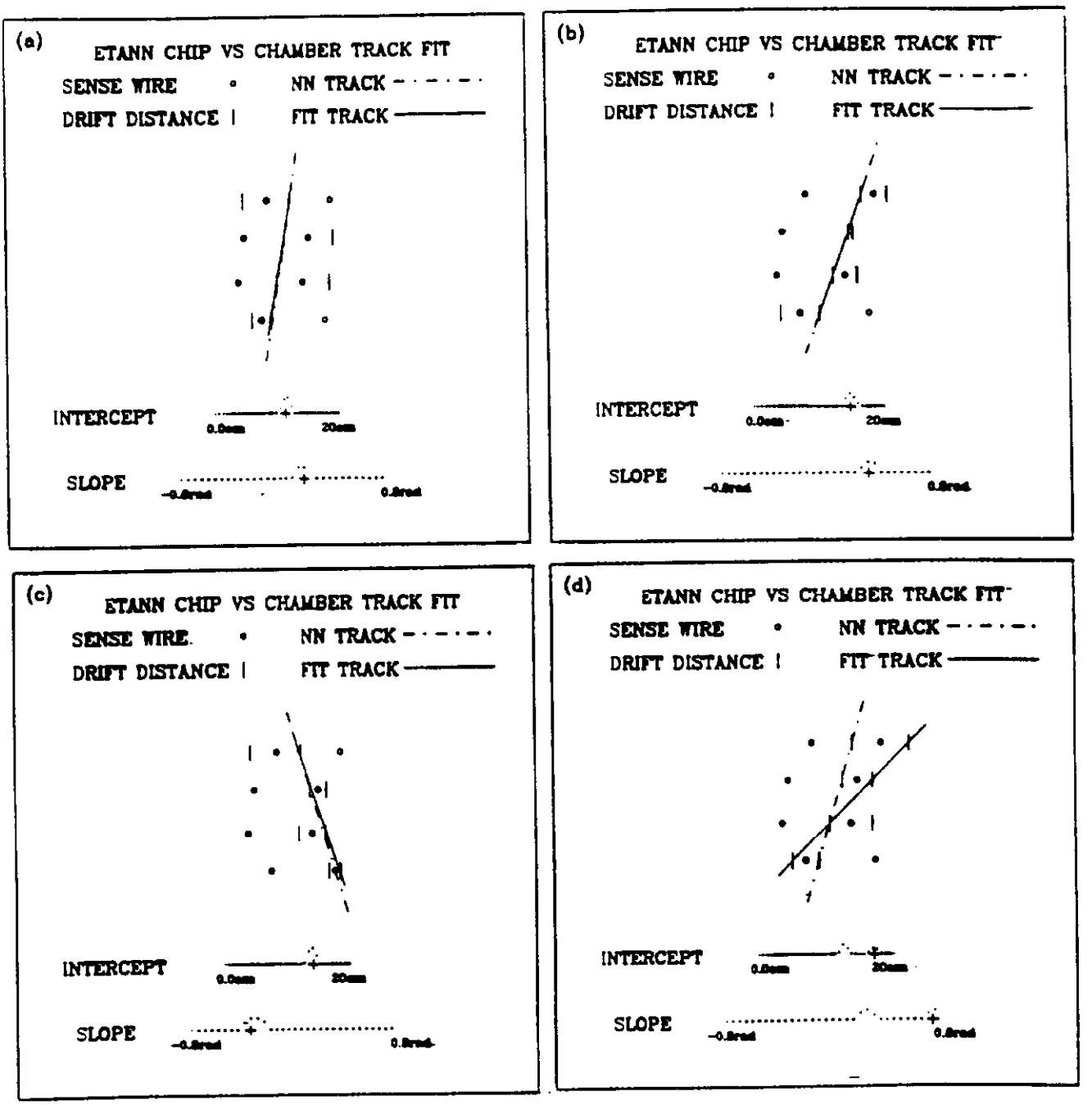

Figure 5: (a,b,c) Cosmic ray events in 4-layer chamber showing the track fit and the corresponding neural network track. Below each track picture is shown the corresponding values of the output distributions of the 32 intercept neurons and the 32 slope neurons. Also shown are fit values with + signs. (d) A case where the neural network chose a different track than the best fit.

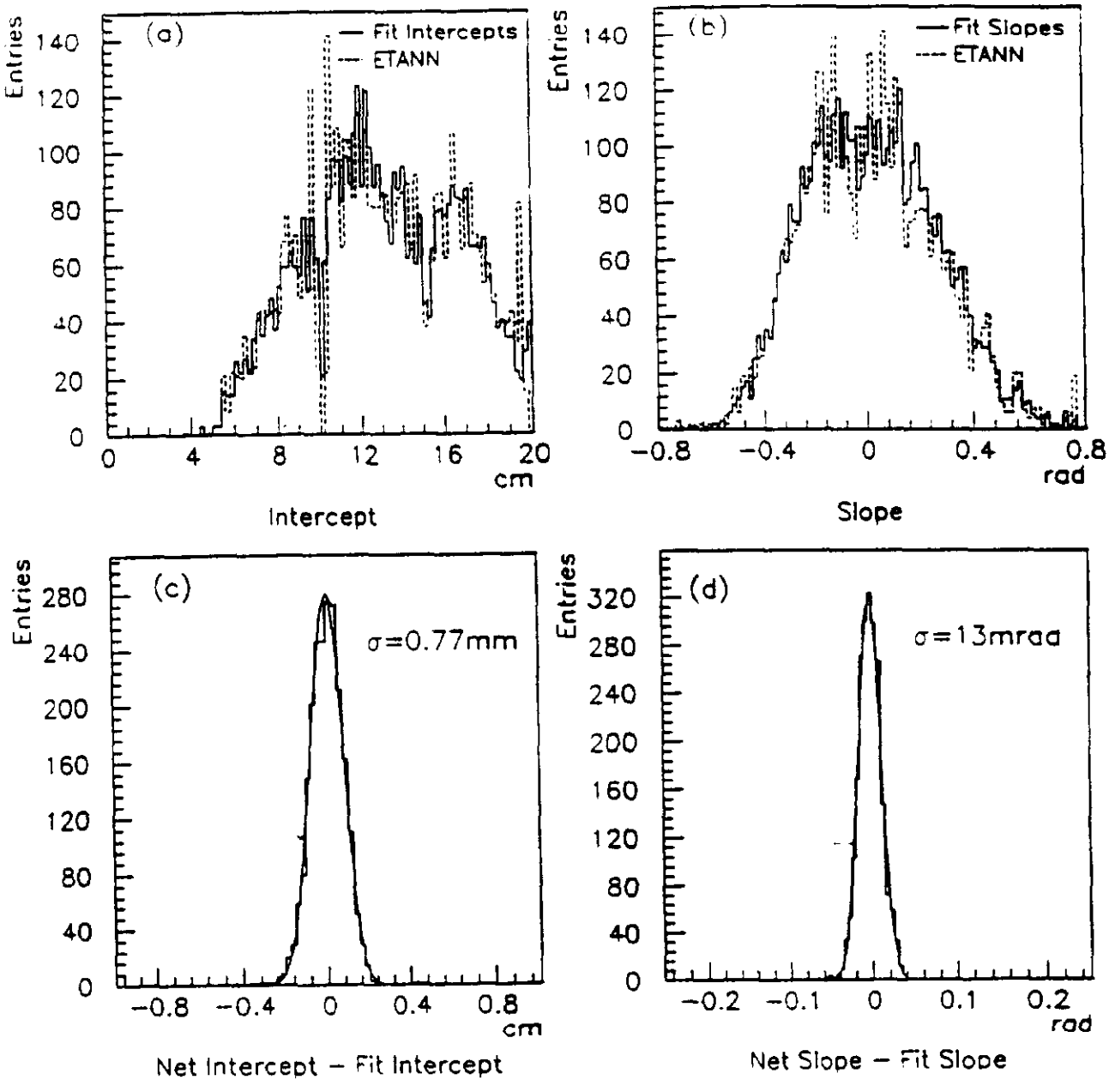

Figure 6: Distributions of (a) intercepts and (b) slopes for both fit and distributed neural network. Distributions of differences in fit and neural net (c) intercepts and (d) slopes, with Gaussian fits.

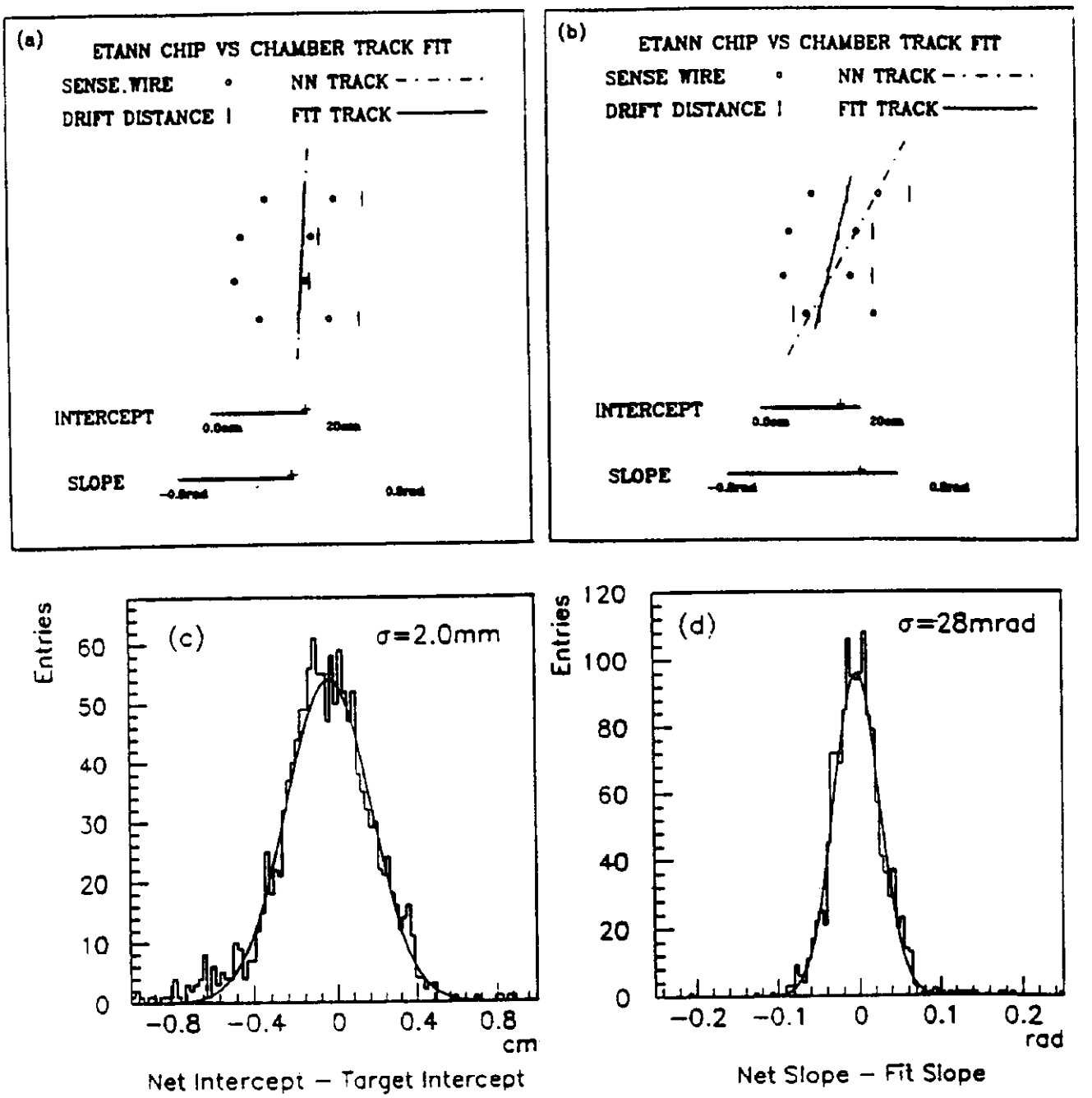

Figure 7: (a)-(b) Two cosmic ray events with the fit track and the proportional net (see fig.4) track. Below each track picture are shown horizontal bars proportional to sum of each set of 10 outputs for intercept and slope, + signs represents fit values. Distributions of differences in fit and neural net are shown for (c) intercepts and (d) slopes, with fits to Gaussians.

possible tracks. Figure 7c-d show distributions of differences between the network values and the fit values for the intercept and slope parameters. The resolutions are 2.0mm (2.4mm) and 28mrad 30mrad with (without) the chi-squared cut.

6 Discussion

Following on a previous study of 3-layer muon chamber tracking with the ETANN chip[3], we have here presented results on tracking with a 4-layer muon chamber. In addition to the distributed output method used before, we have also tried a proportional output type network. The latter gives only about 2.0mm position resolution compared to 0.76mm for the distributed network but it would allow for a much simpler implementation in practice.

The current D0 muon trigger has an effective resolution of about 5cm. Basically it uses the chamber as a hodoscope by looking for pairs of hits in adjacent cells and does not use drift time information[12]. The muon trigger is in the 2nd level of the D0 trigger system and must perform in several microseconds. If the neural networks shown here were incorporated into an upgraded muon trigger system, they would provide a great improvement in the track resolution.

The neural network muon trigger strategy described in ref. 12 would require many neural networks each assigned to a section of muon chamber. Where the occupancy rate is low, signals could be ganged together so as to use a single network for many wires. The outputs of networks from the three chambers of the central muon chamber would be presented to a second level of networks. The second level networks could be trained, for example, to give an output proportional to the momentum of a track or its extrapolated distance of closest approach to the interaction point.

For some applications of neural network tracking one may not need the full power of the ETANN. If a set of weights are found satisfactory, one could implement the network into a circuit which has fixed weights made from simple resistors as discussed in ref. 14. Such nets would be cheaper and simpler than the ETANN especially if needed in large numbers. Also, the ETANN may be too slow for some applications whereas these nets could perform in much less than a microsecond. As a prototyping tool one could use the ETANN for finding the optimum network architecture (e.g. minimum number hidden units needed, best type of output architecture, etc.) and then use a resistor network for the final design.

Acknowledgements

We give our appreciation to Gustavo Cancelo and Sten Hansen of Fermilab for their assistance. We thank Mark Holler, Finn Martin and Simon Tam of Intel for their help. Thanks also to Giovanni Pauletta of the University of Udine and Ken Johns of the University of Arizona. This research was supported by Fermi National Accelerator Lab.

References

[1] B. Denby *Tutorial on Neural Network Applications in High Energy Physics: A 1992 Perspective*, to be published in the proceedings of the Second International Workshop on Software Engineering, Artificial Intelligence, and Expert Systems for High Energy and Nuclear Physics, La Londe les Maures, France in January 1992.

[2] B. Denby, *Computer Phys, Commun*, 49 (1988) 429.

[3] C. S. Lindsey, B. Denby, H. Haggerty, and K. Johns, *Nucl. Inst. & Meth.*, A317 (1992) 346-356.

[4] M. Holler et al.,*Proc. Int. Joint Conf. on Neural Networks*, Washington, D.C., (1989), vol. II, IEEE Catalog 89CH2756, p. 191.

[5] Data booklet for Intel 80170NX *Electrically Trainable Analog Neural Network*, Intel Corp., June. 1991.

[6] Intel Corp., 2250 Mission College Boulevard, MS SC9-40, Santa Clara, Ca. 95052-8125

[7] iDynamind User's Guide, NeuroDynamiX, Boulder Colorado, 1991.

[8] D. Green et al., *Nucl. Inst. & Meth.* A256 (1987) 305.

[9] C. Brown et al., *Nucl. Inst. & Meth.* A270 (1989) 331.

[10] C. S. Lindsey and B. Denby, Nucl. Inst. & Meth., A302 (1991) 217.

[11] T. Akkila, T. Lindblad, B. Lund-Jensen, G. Szekely, and Åge Eide, preprint *A Hardware Implementation of an Analog Neural Network for Gaussian Peak-fitting*, submitted to Nucl. Inst. & Meth. Section A.

[12] M. Fortner, *Analog Neural Networks in an Upgraded Muon Trigger for the D0 Detector*, to be published in Proceedings of the Second International Workshop on Software Engineering, Artificial Intelligence, and Expert Systems for High Energy and Nuclear Physics, La Londe les Maures, France in January 1992.

[13] H. Haggerty, *Discrete Component Hardware Neural Net for Drift Chamber Track Finding*, submitted to NSS Conference Record, 1992 IEEE Nuclear Science Symposium.