

Fermi National Accelerator Laboratory

FERMILAB-Pub-92/55

Real Time Track Finding in a Drift Chamber with a VLSI Neural Network

C. Lindsey, B. Denby and H. Haggerty

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

K. Johns

*University of Arizona, Dept. of Physics
Tucson, Arizona 85721*

February 1992

Submitted to Nuclear Instruments and Methods A.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Real Time Track Finding in a Drift Chamber with a VLSI Neural Network

Clark S. Lindsey^a, Bruce Denby^a, Herman Haggerty^a, Ken Johns^b

^a *Fermi National Accelerator Laboratory **

P.O. Box 500

Batavia, Illinois 60510 USA

^b *University of Arizona, Dept of Physics, Tucson, Arizona 85721, U.S.A.*

ABSTRACT

In a test setup, a hardware neural network determined track parameters of charged particles traversing a drift chamber. Voltages proportional to the drift times in 6 cells of the 3-layer chamber were inputs to the Intel ETANN neural network chip which had been trained to give the slope and intercept of tracks. We compare network track parameters to those obtained from off-line track fits. To our knowledge this is the first online application of a VLSI neural network to a high energy physics detector. This test explored the potential of the chip and the practical problems of using it in a real world setting. We compare chip performance to a neural network simulation on a conventional computer. We discuss possible applications of the chip in high energy physics detector triggers.

1. Introduction

Determining track parameters from drift chamber signals with feed-forward neural networks has been discussed previously.¹⁻³ Neural networks have also been studied for other high energy physics detector applications such as particle identification in Čerenkov counters,⁴ jet identification and background rejection with calorimeters,⁵⁻⁷ and vertex finding in silicon detectors.⁸ Those studies were done with neural network simulations on conventional computers and generally performed satisfactorily. The goal, however, for most of these efforts is eventually to use actual *hardware* neural networks in an on-line environment. Hardware networks should be fast because of their highly parallel architecture and therefore useful in the trigger systems of high energy physics experiments. Recently, neural networks in VLSI chips became available. These hardware neural networks, some of which are digital and others analog, have various limitations and tradeoffs which do not occur with simulations. Depending on the particular chip, there are limits on the numbers of neurons and synapses, on precision of the weights, in the magnitude of the maximum weight

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

values, in maximum gains of the neurons, in processing speeds, etc. Here we give a report on initial test results of one such chip in finding tracks from drift chamber signals.

2. Drift Chamber

Figure 1 shows schematic drawings of the drift chamber used for this study. The chamber is a prototype of the wide angle muon chambers for the D0 detector at the Tevatron.^{9,10} There are both 3 layer and 4 layer chambers in the D0 muon chamber system. The prototype here had 4 layers but the extra layer was ignored for these tests to study tracking with only three layers.

The sense wires are inside of extruded aluminum channels. Figure 1(a) shows a single cell. The cells are 10.15 cm wide. The wall thickness is 0.25 cm. The maximum drift distance in the plane of the sense wires is therefore 4.97 cm. Due to the trigger timing used for the test here, there was some additional dead space resulting in an effective maximum drift distance of about 4.50 cm. Figure 1(b) shows a cross-section of the 6 cells of the prototype chamber. (Actual three layer D0 muon chambers have up to 72 cells.) The 3 cells on the right hand side in figure 1(b) were similar to the cells at the side of a large chamber in that the right hand walls were flush rather than staggered as in all other cells. The dashed lines show the position of the walls for normal cells. The different wall positions causes the time to distance relationship for the right side of these cells to differ somewhat from that for normal cells.

As shown in figure 1(c), the sense wires are connected in pairs. The sense wire signal from one cell goes to the *start* input of a time to voltage converter (TVC) whose stop comes from an external trigger signal. This TVC signal provides the drift time. The output of the same wire also goes to the start of another converter (dTVC, not shown) whose stop comes from the other cell signal. This dTVC signal provides the signal transit time to indicate the cell and the position along the wire where the ionization occurred. A further improvement in this longitudinal measurement comes from signals in metal pads above and below the wires (see figure 1(a)) where signals are induced on the pads by the ionization avalanche at the wire. These pad signals also set latches.

The drift resolution of the chamber in this test is about $500\mu m$. The drift velocity is roughly linear (about $5cm/\mu s$) over the distance from the sense wire to the walls on each side. The fitting algorithm assumes the drift distance represented the distance from the sense wire to the crossing point of the track in the sense wire plane rather than to the distance of closest approach. No attempt here was made to determine the track parameters in the direction along the wire by using the dTVC or pad analog voltages. Instead, the pad latch voltages simply indicate which cell of the pair contained the hit.

3. Neural Network Architecture

Figure 2 illustrates the network architecture for determining track parameters from the drift chamber signals. The three TVC voltages and three pad latch voltages

(from the right hand cells as seen in figure 1b) are inputs to the net. The set of three TVC voltage inputs are actually repeated four times here to overcome the limitation of the maximum weight of 2.5 for the chip (see section 4). There are 64 hidden and 64 output units. The output units are divided into two sets of 32 units, with one set providing the intercept of the track in the plane of the middle layer of wires and the other set giving the slope of the track. Each output neuron corresponds to a *bin* in intercept or slope. The intercept neurons span 20 cm in 0.625 cm bins. In figure 1b the 20 cm spans the range from the left wall of the middle layer to the right dashed line (ignoring the extra drift distance for the edge cell). The slope (given in terms of angle from vertical as shown in figure 1) ranges from -0.8 rad (-45°) to $+0.8$ rad (45°) in bins of 50 mrad (2.9°). The net is trained to excite only those output neurons which correspond to the slope and intercept of the track. The target patterns are given as Gaussian distributions ($\sigma = 0.5$ cm for intercept and 40 mrad for slope) across 3 or 4 neurons so that the slope and intercepts can be calculated more precisely than the width of a single bin (see figure 2). Finding the bin with the maximum output and then calculating the average over ± 2 bins of the maximum gives the network intercept and slope.

An alternative output architecture would have two output neurons, one of whose activation is proportional to the intercept position and the other's activation proportional to the slope. However, the jitter in the electronics for the chip used here causes a percent or so variation in the output of a given neuron for repeated presentation of the same input pattern. This alone would give about factor of two worse intercept resolution than what was achieved with the multiple output distribution method used here. The averaging around the maximum output neuron reduces the effect of the jitter. Also, for cases with ambiguities where more than one possible track choice is possible (at least within the *resolution* of the network), there can be activations for more than one intercept or slope. This ability to provide secondary answers could be useful if, for example, a trigger system combined the network output with other tracking information, perhaps from another chamber. The secondary answer might be found to be more consistent with this other information than the primary answer. (Reference 2 discusses a simulation where multiple network outputs were combined to find the event vertex.)

4. Intel ETANN Chip

The Intel ETANN (Electrically Trainable Analog Neural Network) chip¹²⁻¹⁴ uses EEPROM technology to make modifiable, but non-volatile, synapses or weights. Two floating gate pairs make up a synapse differential amplifier (similar to a Gilbert multiplier circuit) such that the relative thresholds between the gates determines the output current for a given input voltage. Altering the charge on the floating gates alters these thresholds. The floating gates thereby provide the variable weights. With respect to a given reference voltage, both inputs and weights can be positive or negative and so provide full 4-quadrant multiplication. The effective precision of the weights are 6-7 bits and range between ± 2.5 in magnitude. The input voltage can

be from 0.0v to 3.5v. The chip has 64 inputs and 64 neurons, plus 16 bias voltages. There are two sets of 80x64 weight arrays. The 64 input voltage signal *vector* makes dot products with each column of the first weight set and the resulting current sums feed into the corresponding neuron, which consists of a threshold amplifier with a sigmoidal transfer characteristic. The sigmoid transfer function (for reference voltage of 1.6v) is roughly:

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

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 the input voltage for the first layer or the output of the first layer neurons going into the second layer, and w_{jk} is the weight for the connection between units j and k . Seven bias neurons of voltage $V_b \simeq 4.0v$ are provided. The gain G can be varied¹⁵ but here was set near the maximum of about 1.0.

After the first layer processing, the neuron outputs can be clocked back through the second weight set to get two layer performance. So a single ETANN can implement a net with up to 64 inputs, 64 hidden units, and 64 output units. (An option is to clock in two sets of 64 inputs to make a net with 128 inputs and 64 output units.) The processing time depends somewhat on the number of inputs and neurons used. The first layer processing takes a maximum of $3\mu s$ and the second layer a maximum of $5\mu s$ for a total of $8\mu s$ for 2 layer performance.

In addition to the chip, a PC based development system¹² is available. The chip is connected to an external *trainer* box which provides communication between the PC and the chip and can also generate the large voltage pulses needed to alter the floating gate charges. With the trainer we can (1) test and initialize the chip; (2) emulate the chip; (3) write or read weights to and from the chip; (4) do chip-in-the-loop (CIL) learning. The latter is required because the chip does not have a built-in learning algorithm. For CIL, a voltage pattern is presented to the chip, the chip output voltages digitized and compared by the PC to target outputs. The PC then does back-propagation using the emulation sigmoids and derivatives to obtain the necessary weight changes. The new weights are then down-loaded to the chip. This process is repeated over many patterns until the average difference in the outputs and targets is small enough to judge that the chip has *learned* the training set.

5. Chip Training and Testing

To train the chip the following procedure was carried out. Training sets of 10000 track patterns were made with a simple Monte Carlo simulation of the chamber. Monte Carlo tracks traversed the cells at random intercepts and slopes. The drift distances (distance to the sense wire in the plane of the wires) for a given track were smeared with a Gaussian of $500\mu m$ sigma to simulate the chamber drift resolution. Using these smeared drift distances, the target slopes and intercepts were calculated with a least squares fit. The drift distances were converted to TVC voltage values.

Each pattern consisted of the 3 input TVC voltages (repeated 4 times, see figure 2) and 3 latch pad voltages for a total of 15 input values, plus 64 target values (32 for intercept and 32 for slope, see figure 2) for the output neurons. The TVC voltage is 3.2v if the track passes right at the sense wire and is 0.0v for the maximum drift distance of 4.5cm. The pad latch voltage is 0.1v if the track is in a left hand cell (as seen in figure 1b) and is 3v if track is in a right hand cell. Output target voltages were set from 0.1v to 3.0v.

A back-propagation¹¹ program (the standard back-propagation algorithm was modified to limit weights to a maximum of 2.5 in magnitude) processed the training set for several million iterations on a fast workstation. The resulting set of weights was then downloaded to the PC emulator. The emulator does a better simulation of the chip but the PC is much slower and the number of patterns that can be stored in memory is smaller. The emulator, in stand-alone mode, did back-propagation for several hundred thousand iterations on several files of 2000 patterns each. Finally, the emulation weights were downloaded into the chip. Chip-in-the-loop (CIL) back-propagation was then carried out for a few thousand iterations on several sets of 600 patterns. It is advantageous to do the preliminary training in software because the CIL training takes several hours to approach the performance of the emulator (even though the initial weights are from the emulator) and because the manufacturer recommends minimizing the number of CIL iterations. Apparently, the large voltage pulses required to alter the floating charge on the gates eventually degrade their performance.

With the chip connected to the trainer, we present new pattern sets not used in the training to test how well the chip generalizes. Figures 3 through 6 show results of tests with the chip in the trainer. Figure 3 illustrates four events presented to the chip. Each frame shows the target track and the network track and the activations of the output neurons. The target parameters are shown as plus signs on the same distributions. Figures 4a and 4b show distributions of the target intercepts and slopes of tracks as compared to what the chip gives. The intercept distribution was generated roughly flat across the 20 cm range. The gap in the middle of the distribution in figure 4(a) is due to the dead region at the wall separating the cells. The slope distribution is not flat because of the staggering of the layers and the requirement that one cell of each pair of each layer have a hit. Figure 4c and 4d show distributions of the differences in the target and chip intercepts and slopes. Also, shown are the same distributions for the workstation simulation. Fitting to Gaussians, the workstation simulation sigmas are 0.51 mm and 6 mrad (0.34°) for intercept and slope respectively, whereas for the chip the values are 0.90 mm and 16 mrad (0.92°). This indicates that the limited precision of the chip, as well as the inability to train on large pattern sets for millions of iterations, does degrade the performance compared to a simulation. The workstation simulation had the requirements of maximum weights ± 2.5 and gain of 1.0, but it had the precision of 32 bit floating point numbers and the arithmetic operations were exact. The chip has about 6-7 bit precision plus non-linearities in the the input-weight multiplications.

The network does not always find the correct track. This is caused primarily by patterns with ambiguous tracks, which occur easily for a three layer chamber despite the staggering of the cells. Figure 5 shows four examples of such ambiguous patterns. By eye, in each of the examples, there are two possible tracks. The least square fit calculation found the smallest chi-square for tracks different from the ones chosen by the net. This indicates that the *resolution* of the net is not quite precise enough always to find the best choice. Scanning of such events reveals that usually there is some activation at the slope and intercepts neurons of the best fit track but the largest activations were for the phantom track parameters.

To a lesser extent, this inefficiency also occurs in the workstation simulation where the low maximum weight and gain seems to limit the accuracy. Figure 6 shows track finding efficiencies for both the simulation and the chip in the trainer. Figure 6a shows the efficiency of the network having the correct intercept within 5 mm of the target intercept as a function of target intercept. This chip intercept efficiency averages about 96% across the cell compared to 98% for the simulation. Figure 6b shows the efficiency in finding the slope within 100 mrad (5.7°) of the target slope as a function of slope. The chip average efficiency is about 95% versus 97% for the simulation. Figures 6c and 6d show the combined efficiency of simultaneously requiring intercept within 5 mm and slope within 100 mrad of the target as function of intercept and slope. The combined efficiency for the chip is about 93% versus nearly 97% for the simulation. The biggest dips in figures 6a and 6c occur in the intercept ranges where ambiguities occur as shown by examples in figure 5.

6. Drift Chamber Test Setup

Figure 7 shows a schematic of the test setup at Fermilab. The test was run in parasitic mode by placing the chamber near a beam absorber of another experiment to obtain muons from particle decays. A trigger required hits in scintillators placed behind the chamber plus a coincidence of one hit from each layer of the chamber. For cosmic ray tests the chambers and scintillators were rotated 90 degrees. The analog TVC voltages (0.0v to 3.2v) and pad latch voltages (0.1v and 3.0v) were sent both to the ETANN inputs and to analog-to-digital converters (ADC). For two layer processing, the ETANN is sent pulses to clock the first layer neuron outputs back through the second set of weights. The second layer outputs are connected to ADC's and read out along with the input digitization. The data (each event record included ADC values for 3 TVC's, 3 pad latches, and 64 ETANN outputs) was recorded on disk for later processing. The ETANN resided on a VME board in a crate adjacent to the chamber.

7. Drift Chamber Test Results

Figure 8 shows some early results of on-line track finding with the ETANN connected to the chamber while the muon beam was available. Figures 8a and 8b show distributions of the slopes and intercepts for the beam tracks for the fits and for the ETANN outputs. The slope distribution differs from that in figure 4 because of the narrow beam. The differences between the net and fit intercepts and slopes

are shown in figure 8c and 8d. The Gaussian fits give about 1.2 mm resolution for the intercepts and about 20 mrad (1.1°) for the slope. This compares to 0.9 mm and 16 mrad (0.9°), respectively, for the chip in the trainer. Also, the intercept distribution is shifted slightly negative by about 0.400 mm.

Ideally the chip should give identical results whether connected to the trainer or to the VME board when presented with identical input voltage patterns. To reduce the differences between the performance of the chip with the trainer and with the chamber, several systematic problems were addressed. At the chamber the TVC voltages are roughly on a 0.0v to 4.0v scale. The chip was trained on a 0.0v to 3.2v scale. A voltage divider reduces the TVC voltage from the 4v scale to the 3.2v scale. A careful comparison was done of the voltage going into the chip with the TVC ADC readings to insure that the voltage divider was correct to within 1% and that the pedestal voltage was truly at 0.0v. The chip power supply voltage, reference voltages and gain voltage were measured both at the trainer and on the VME board to insure they were the same to within 10mV. This is very important because, as shown in reference 14, a variation of even a few millivolts in the power supply voltage can cause significant shifts in the neuron sigmoidal outputs. Such shifts in the hidden layer sigmoids in turn can cause dramatic changes in the final output layer sigmoids. For example, a difference of 14mV in the power supply voltages between the trainer and the VME board caused as much as a factor of 2 change in some output neuron values.

While this study of systematics was underway, the Fermilab fixed target run ended. So the chamber and trigger counters were oriented for cosmic ray data taking. Figures 9a and 9b show the distributions of intercepts and slopes for cosmic rays for both the fits and the network outputs. After improved control of systematics, the results became more consistent with the trainer results. The Gaussian fits to the distributions (where net outputs were compared to those tracks with fits having chi-squared per degree of freedom less than 1.0) in figures 9c and 9d give sigmas of 0.85 mm and 15 mrad for intercept and slope, respectively, (0.97 mm and 15 mrad if no cut on chi-squared is made). These widths are similar to what was obtained with the chip in the trainer. Figures 10a and 10b show that, as before, the net is inefficient at the 5-10% level due to the ambiguity problem. The worse inefficiency at large angles in figure 10b as compared to figure 6b is apparently due to a greater number of cases like figure 5d in the cosmic ray distribution of tracks than in the flat intercept distribution of the pattern sets made for training and testing (compare figures 4a and 9a).

To compare further the performance of the chip in the trainer versus that when connected to the chamber, the above cosmic ray data files were made into pattern sets for the trainer. This was done by converting the TVC and pad latch ADC values into trainer input numbers and making the track fit intercepts and slopes into target distributions. The chip was returned to the trainer and the dashed lines in figure 10 show the results for these same data when these patterns were presented to the chip. Ideally there would be no difference between the solid and

dashed distributions. The fact that the differences are fairly small indicate that there are no big systematic problems in chip power supply voltage levels, ADC pedestal subtractions, etc. This provides us the option of training on actual data rather than Monte Carlo generated events, where the latter may be based on an incomplete representation of the chamber.

8. Discussion

This test was done both to determine how well the chip could implement a given neural network and also to find if it could perform a task of possible importance to triggering in high energy physics detectors. Initially it was not known if the limited precision of the weights, low maximum weight, low sigmoid gain, non-linearities, etc. would preclude implementing complicated networks, especially those whose desired outputs were proportional values rather than simply binary (i.e. above or below some threshold). Although the chip is less precise than a simulation on a conventional computer, we believe the results are quite promising and indicate the chip could be a useful tool both for high energy detector systems and for other applications.

We believe this is the first time that a hardware neural network of any kind has been used to do on-line pattern recognition with a high energy particle detector. The neural network learned from examples to find intercepts and slopes of tracks from drift chamber signals to a resolution less than 1mm in intercept and less than 1 degree in slope. There are long tails on the distributions, though, caused by situations where the network cannot resolve ambiguous tracks. This produces inefficiencies and false tracks.

Further studies of the chip and training procedures are being done to try to improve the resolution and the efficiency. In addition we hope to apply the same method to a four layer chamber (see section 2). The three layer chamber maximizes the ambiguity problem. A four layer network should have much higher efficiency because of the elimination of most phantom tracks with the constraints of an extra layer.

This test, however, illustrates that the ETANN chip already reaches a level of performance that is required for some trigger applications. Typically, trigger systems do coarse measurements but they do them very quickly to determine whether to record the event on tape or to reset the system in time for the next beam crossing. There are usually several levels of triggers in current collider experiments. The lowest order level 0 triggers make the simplest cuts and take on the order of $1 - 2\mu s$. Level 1 triggers do somewhat more elaborate cuts and take on the order of $10 - 20\mu s$. Further trigger levels continue the process to reduce tape writing rates. Typically the level 0 and 1 triggers are done in hardware and the higher level triggers in software. The use of hardware neural networks is being investigated to do fast muon track finding in the level 1 trigger for the D0 $p\bar{p}$ collider experiment.¹⁶ The current D0 forward muon trigger system, for example, has an effective resolution of about 5cm compared to the 1 mm resolution achieved here. This improved resolution, even with the inefficiencies mentioned above, would reduce backgrounds considerably.¹⁷

The final design of such a trigger would probably have a network architecture quite different from the net used here. For example, the network probably would have a least 6 cell pair inputs instead of 3 to allow for tracks crossing pair boundaries and allow for overlapping coverage. Also, it is probably only necessary for the chip to give the intercepts of tracks. The performance of the simpler network reported here, though, gives us some initial benchmarks to build upon.

Acknowledgements We thank Mike Fortner of Northern Illinois University for useful discussions on the D0 trigger system. We give our appreciation to Gustavo Canceledo and Sten Hansen of Fermilab for their assistance. Also we thank Mark Holler of Intel. This research was supported by Fermi National Accelerator Lab.

References

1. B. Denby, E. Lessner and C. S. Lindsey *Proc. 1990 Conf. on Computing in High Energy Physics*, Sante Fe, NM, 1990, AIP Conf Proc. **209**, 211.
2. C.S. Lindsey and B. Denby, *Nucl. Inst. & Meth.* **A302** (1991) 217.
3. C.S. Lindsey, "Tracking and Vertex Finding in Drift Chambers with Neural Networks", to be published in proceedings of the "Workshop on Neural Networks: From Biology to High Energy Physics", Elba, Italy, June 1991.
4. T. Francke et al., *Nucl. Instr. and Meth.*, **A307** (1991) 47.
5. D. Cutts et al., *The Use of Neural Networks in the D0 Data Acquisition System*, Real-Time '89 Conference, Williamsburg, Va. (May 1989).
6. B. Denby et al., *IEEE Tran. Nucl. Science*, **37**, No.2, (1990) 248.
7. J. Fent, C. Kiesling, P. Ribarics, *A Level 2 Calorimeter Trigger Using Neural Networks: Status Report*, Experiment H1 internal report H1-MPI-150, 23.4.1991.
8. L. Gupta et al., *Neural Network Trigger Algorithms for Heavy Quark Selection in a Fixed Target Physics Experiment*, FERMILAB-Pub-91/117, submitted to *Pattern Recognition* (May 1991).
9. C. Brown et al., *Nucl. Inst. & Meth.* **A270** (1989) 331.
10. D. Green et al., *Nucl. Inst. & Meth.* **A256** (1987) 305.
11. D. Rumelhart et al., *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, vol. I, chap. 8, (MIT Press, Cambridge, Ma, 1986).
12. Intel Corp., 2250 Mission College Boulevard, MS SC9-40, Santa Clara, Ca. 95052-8125
13. M. Holler et al., *Proc. Int. Joint Conf. on Neural Networks*, Washington, D.C., (1989), vol. II, IEEE Catalog 89CH2756, p. 191.
14. Data booklet for Intel 80170NX *Electrically Trainable Analog Neural Network*, Intel Corp., June. 1991.
15. There is an optional high gain mode of 0.0 to 5.0v output. The gain is about 10 times larger but the transfer function is not suitable for use as a sigmoid. This High gain mode is used for purely binary type nets.
16. D0 Design Report. Dec. 1983.

17. M. Fortner, *Analog Neural Networks in an Upgraded Muon Trigger for the D0 Detector*, presented at 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.

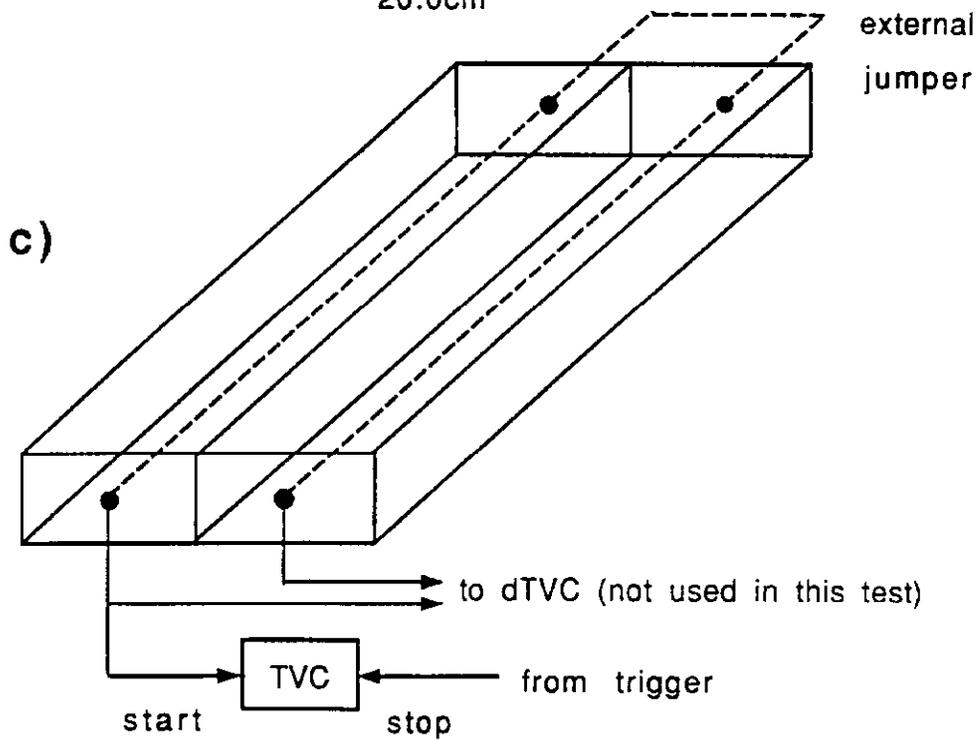
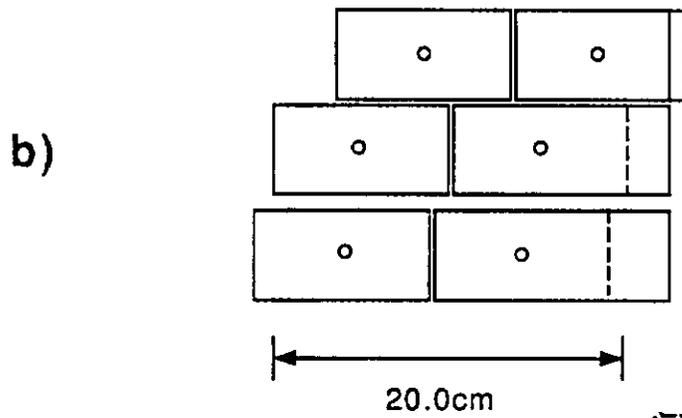
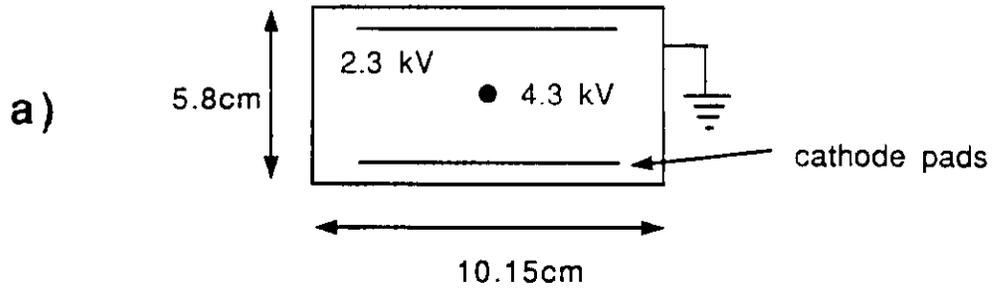


Figure 1. (a) Cross-sectional view of a D0 muon drift chamber cell. (b) The 6 cells of the prototype chamber. The dashed lines indicate position of walls for normal width cells. (c) Top view of two cells showing the electrical connection of the sense wires at one end.

NEURAL NETWORK FOR D0 MUON CHAMBER TRACKING

Input = 4 x (3 TVC Voltages) + 3 Pad Latch Voltages
 Output = 32 0.625cm bins from -0cm to +20cm
 + 32 0.05rad bins from -0.8rad to 0.8rad

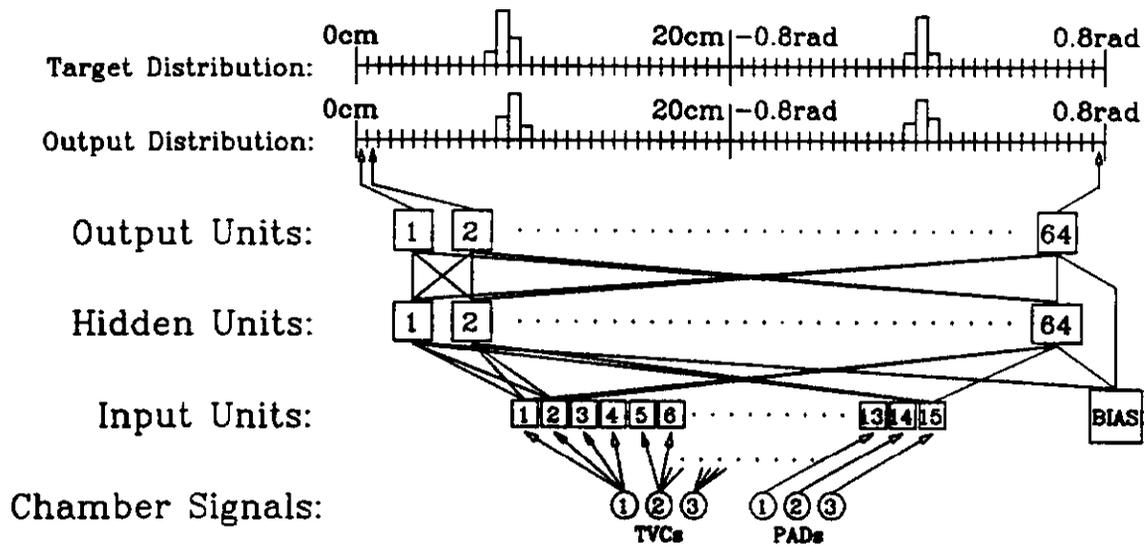


Figure 2. Neural network architecture to determine intercept and slope of tracks from 6 cells of the D0 muon drift chamber prototype.

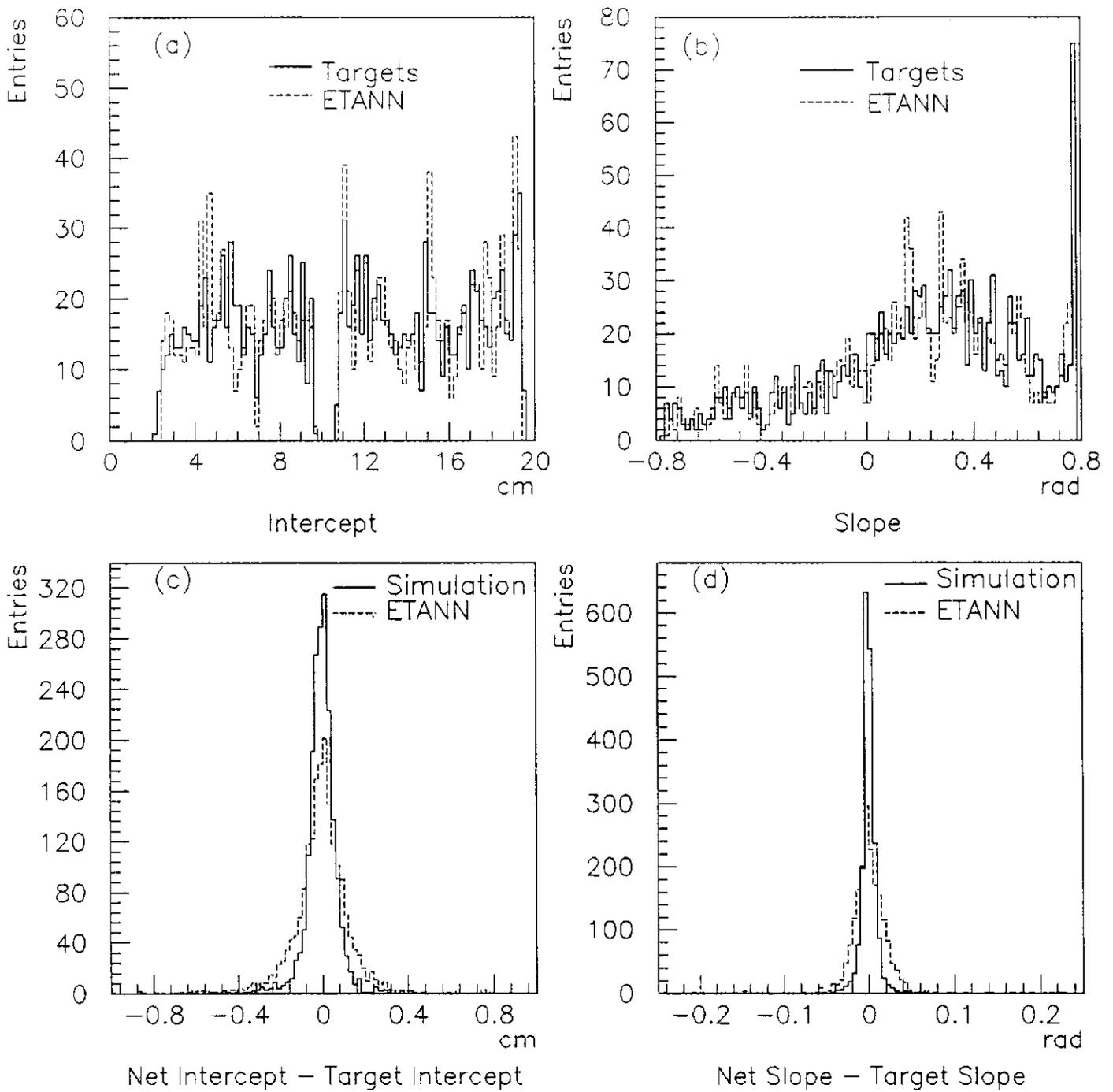


Figure 4. For chip in trainer: (a) Intercepts of targets and neural net; (b) slopes of targets and net; (c) distributions of differences in intercept of target and net; (d) distributions of differences in slopes of target and net. For (c) and (d) shown are results for both chip and the workstation simulation.

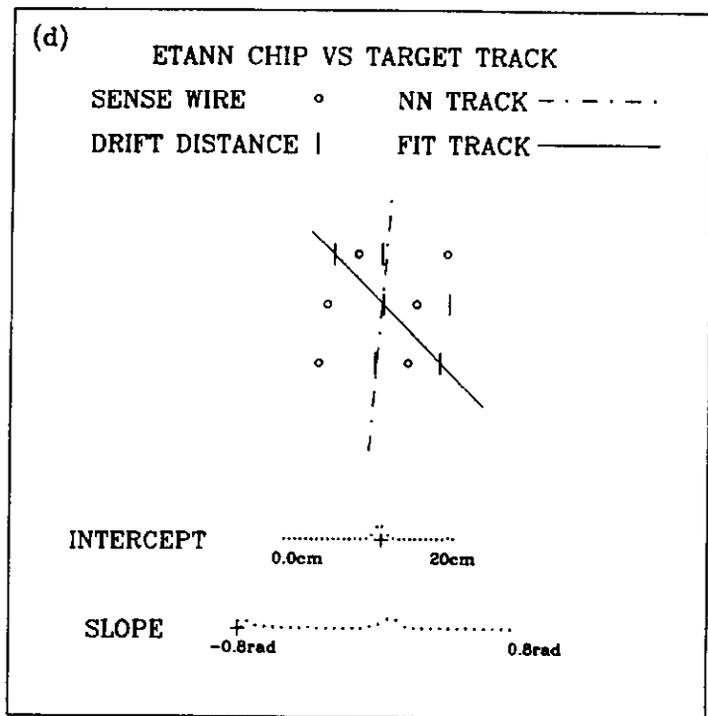
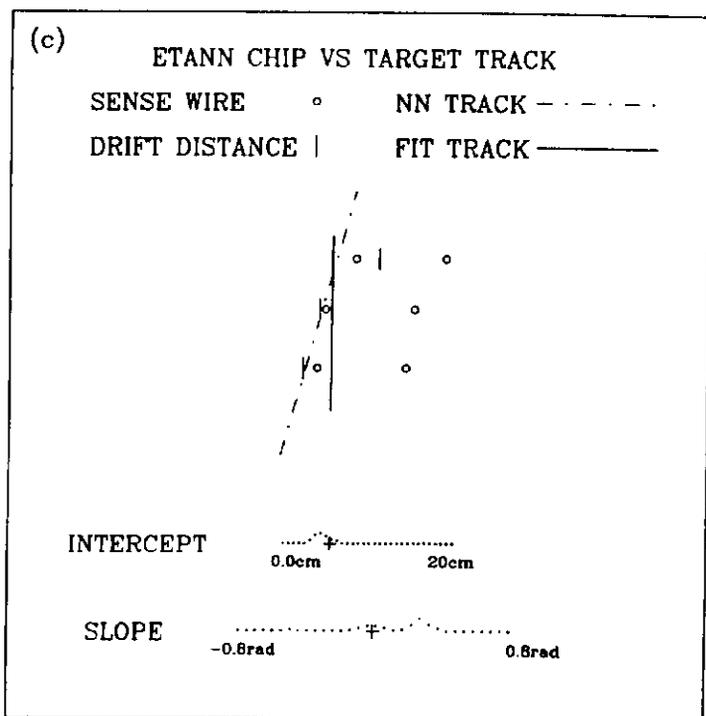
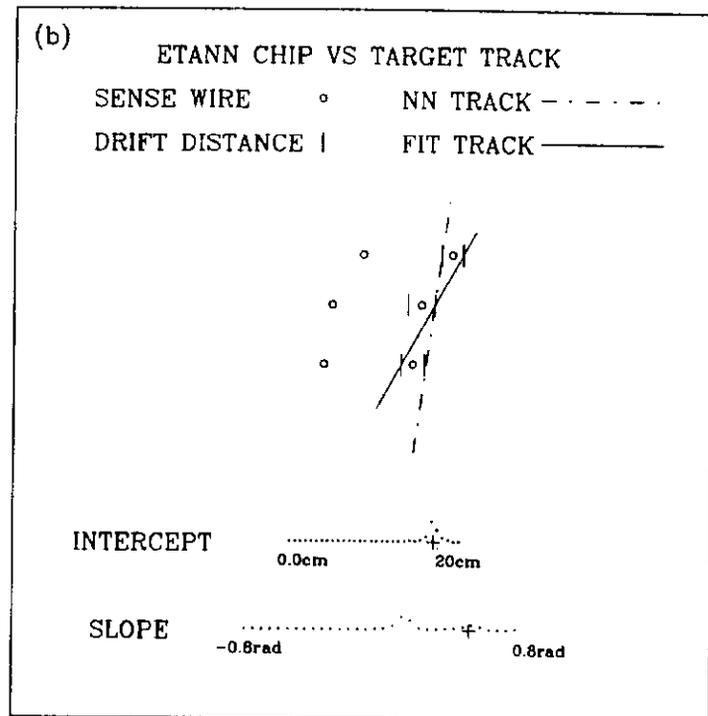
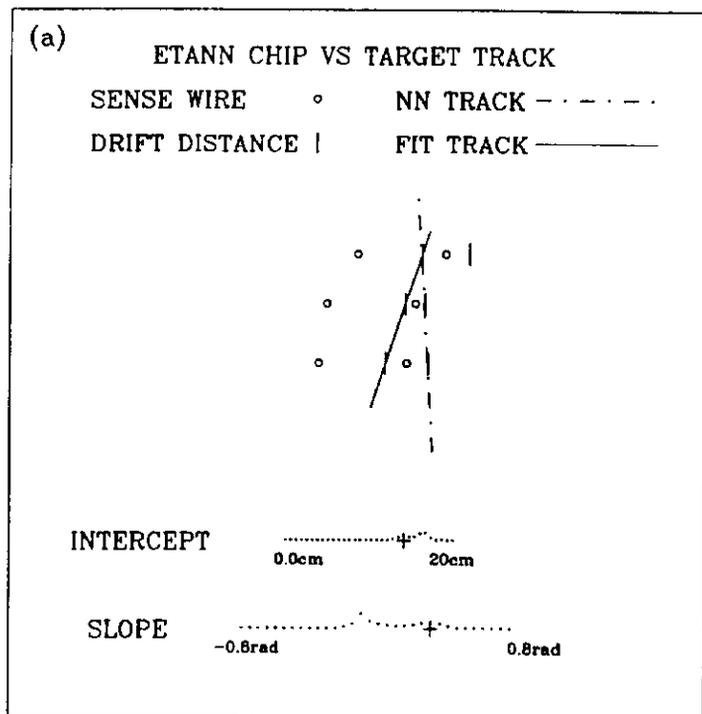


Figure 5. Examples of Monte Carlo track patterns presented to the chip (while connected to the training system) where the chip gave incorrect slope and/or intercept as compared to best fit parameters.

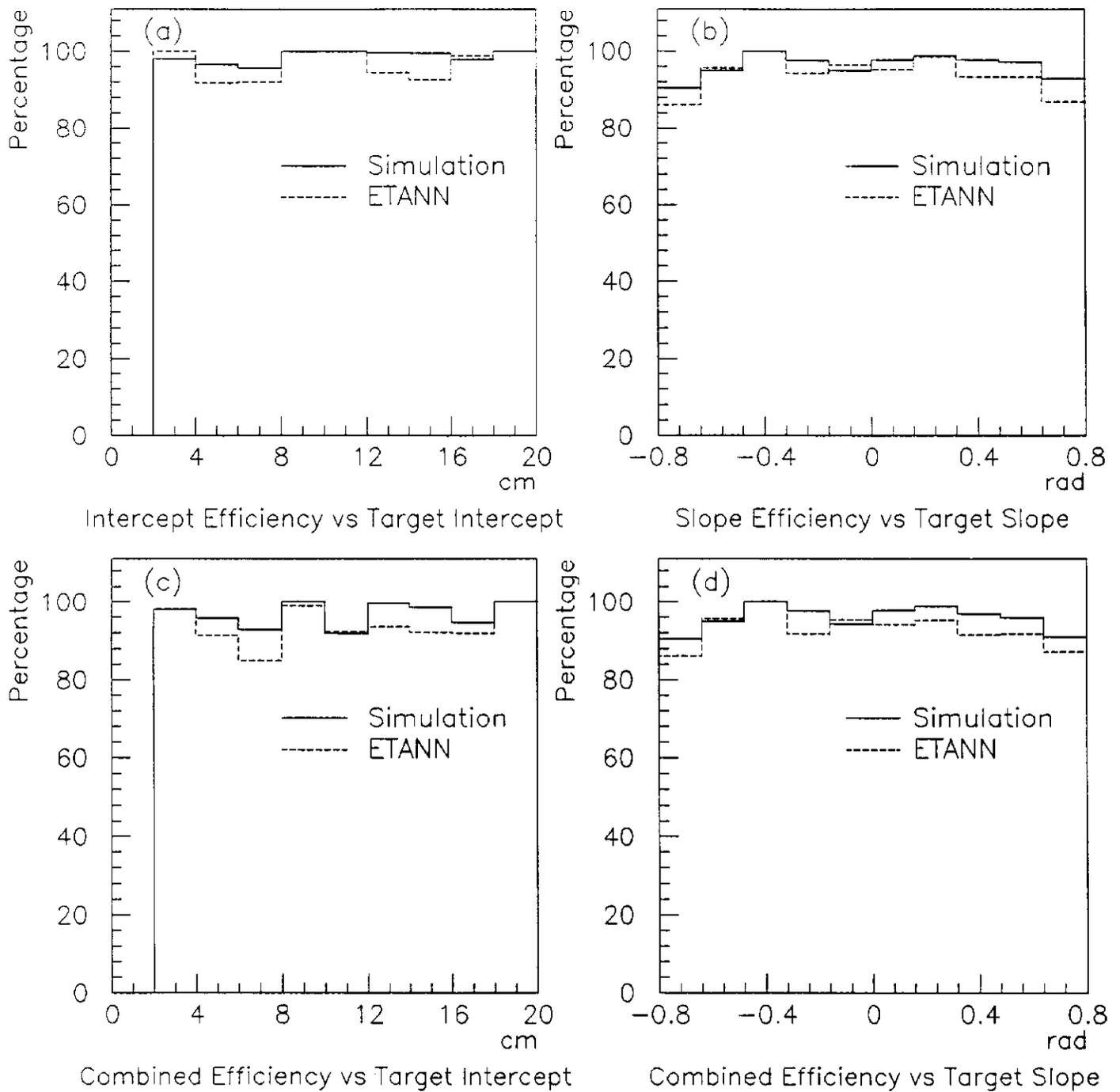


Figure 6. For chip in trainer (a) percent of net intercepts within 5mm of targets as function of target intercept; (b) percent of net slopes within 100 mrad of targets vs target slope; (c) percent of net intercept and slopes within 5 mm and 100 mrad, respectively of target values vs target intercept; (d) same but vs target slope. Shown are result for both chip and workstation simulation.

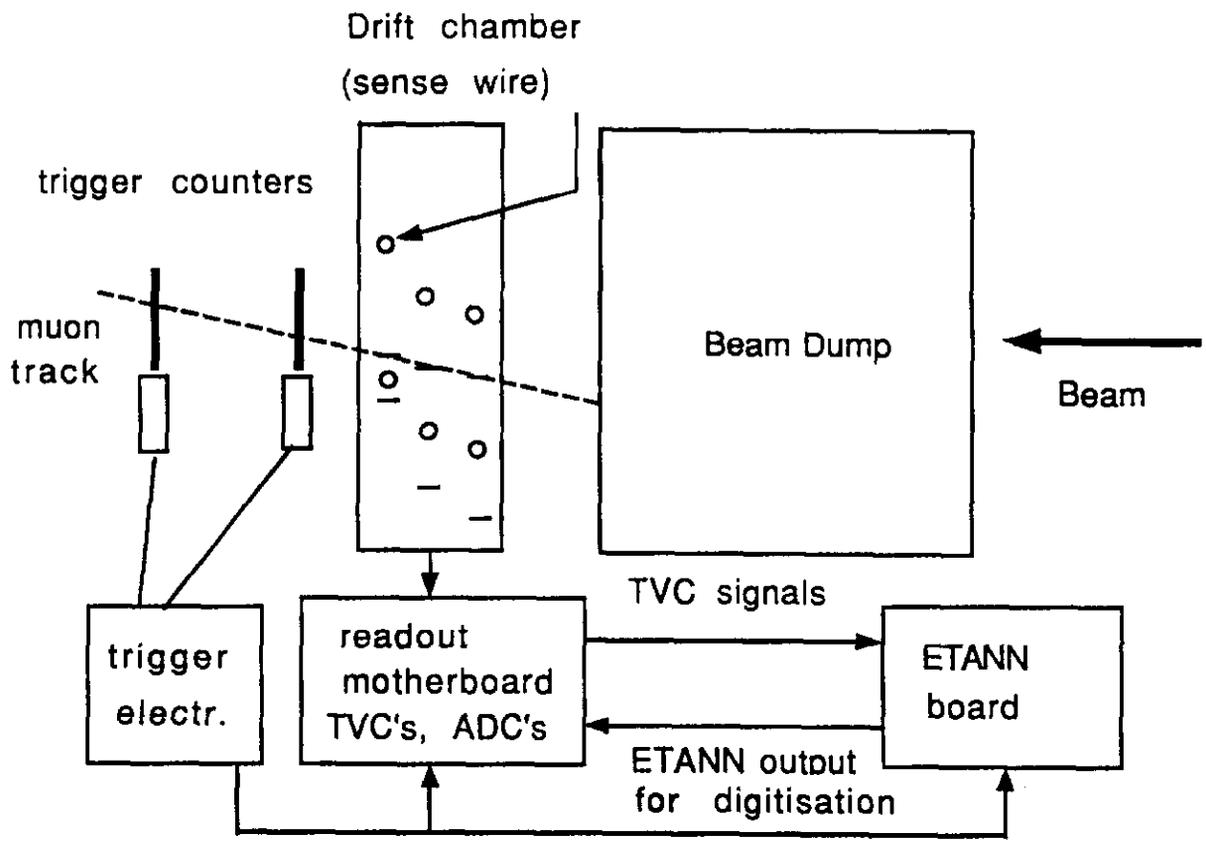


Figure 7. Chamber setup during beam tests. For cosmic ray data taking, the chamber was rotated counter-clockwise 90° and one trigger counter was placed above and one below the chamber. The motherboard TVC and ADC data were read out and saved by a computer not shown.

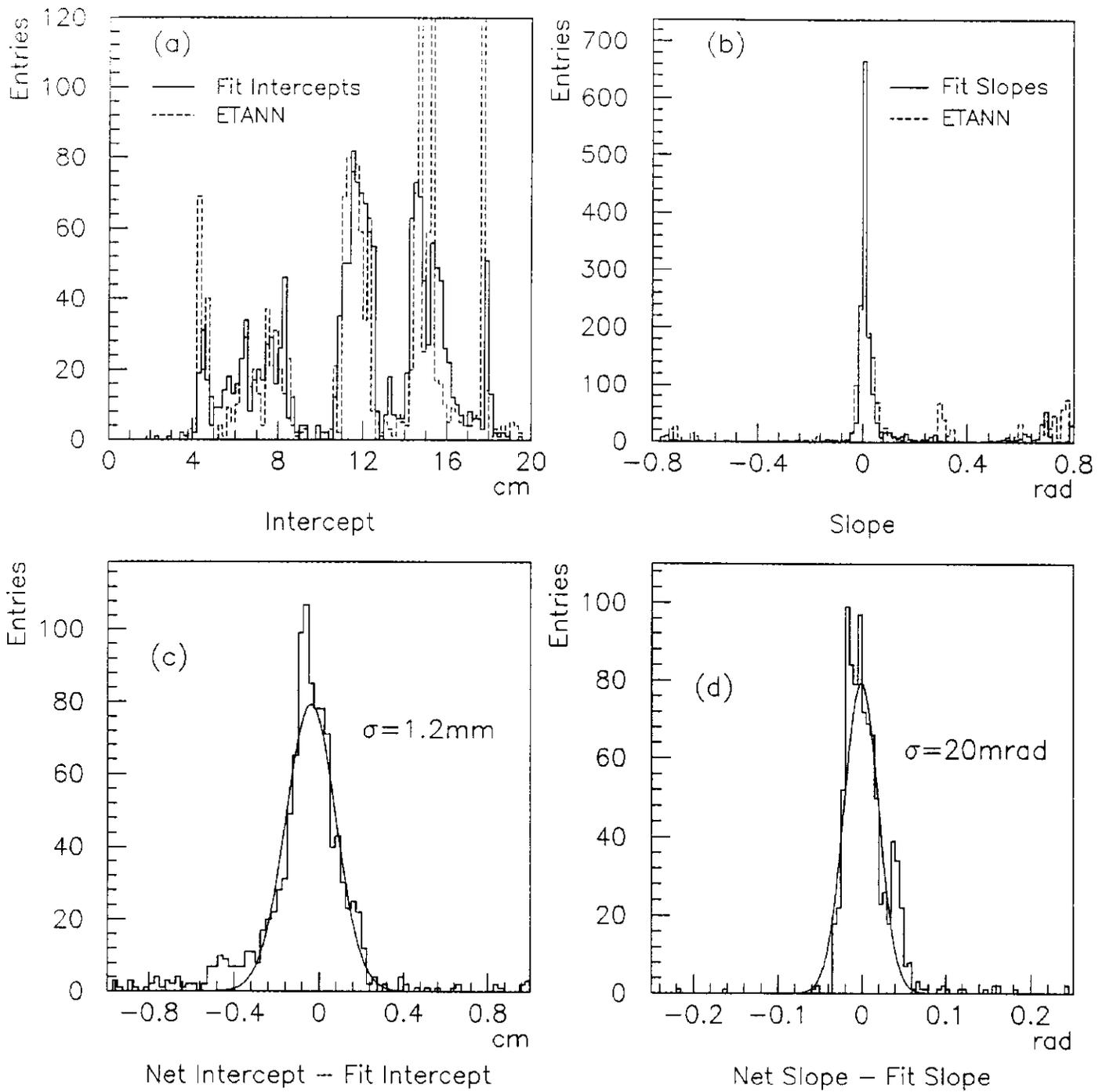


Figure 8. For a sample of beam data: (a) distribution of intercepts for fits and ETANN; (b) slopes; (c) distributions of differences in fit and ETANN intercepts; (d) differences in slopes; (e) differences in intercepts.

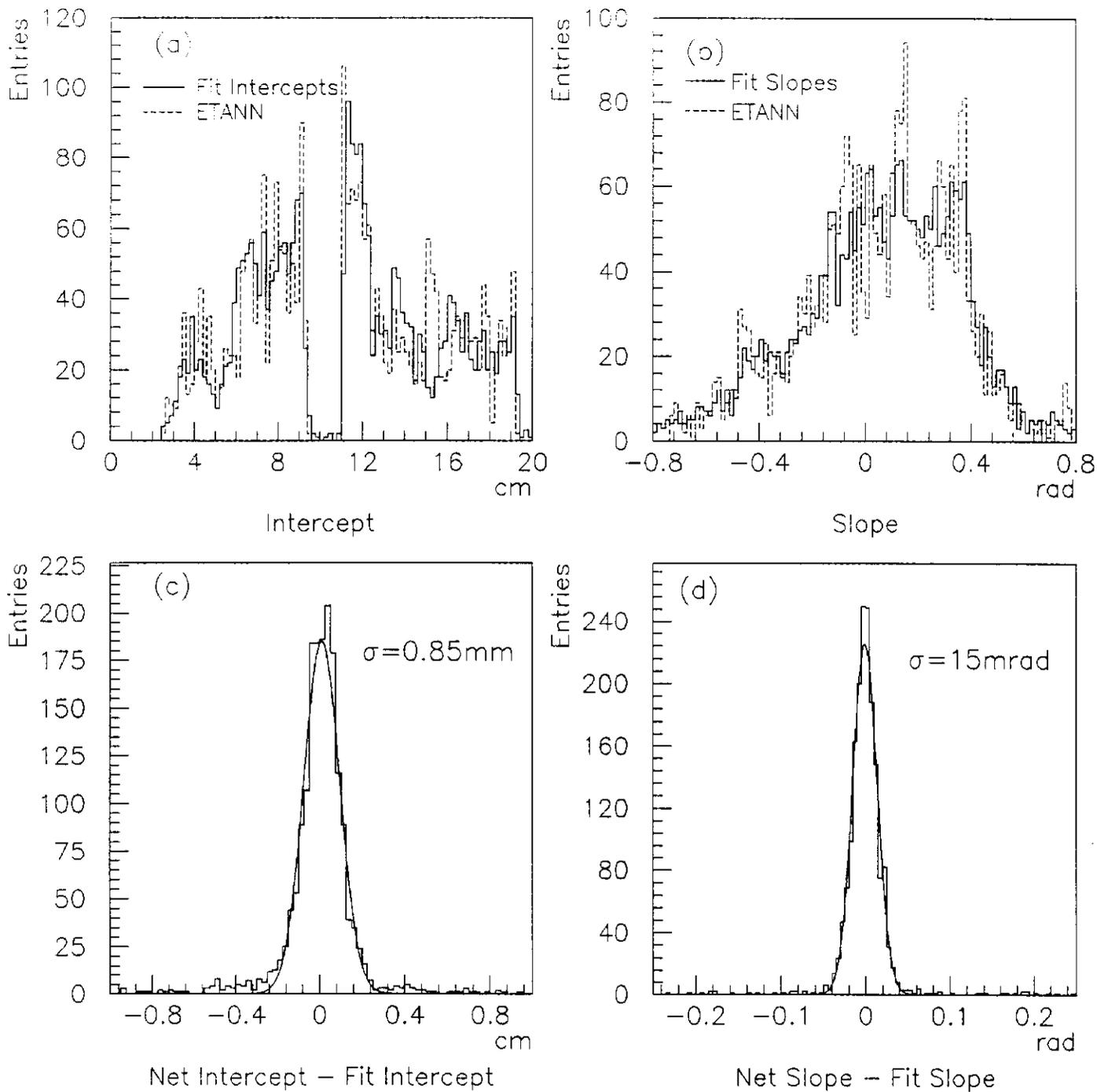


Figure 9. For cosmic ray data: (a) distribution of intercepts for fits and ETANN; (b) slopes; (c) distributions of differences in fit and ETANN intercepts; (d) differences in slopes.

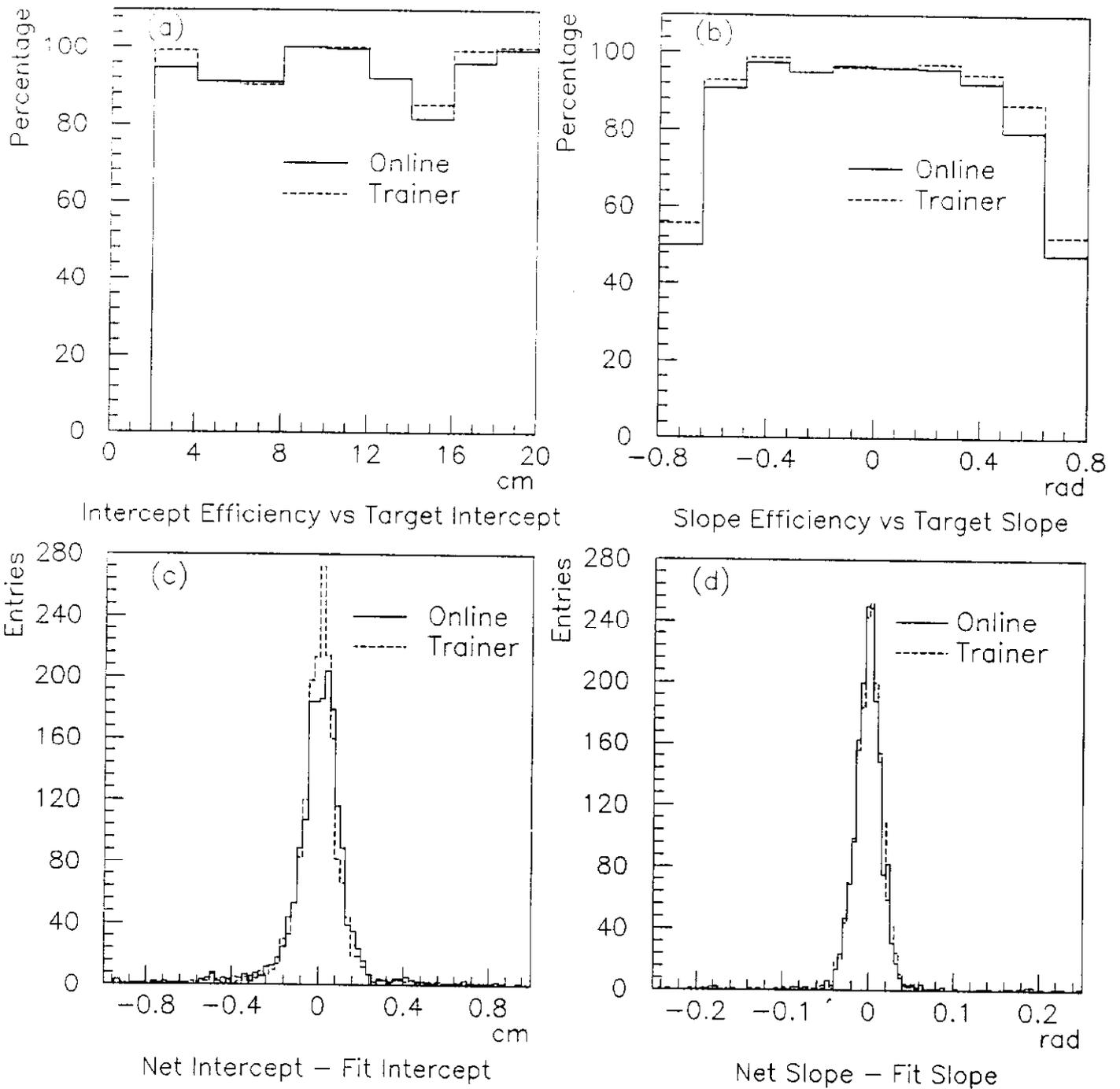


Figure 10. Cosmic ray online results compared to same data made into pattern sets and presented to the chip while in trainer: (a) Intercept efficiency versus fit intercept; (b) Slope efficiency vs fit slope; (c) distributions of differences in fit and ETANN intercepts; (d) differences in slopes.