PATTERN RECOGNITION FOR PHENIX MUON ARMS

D. Dutta, A. K. Mohanty, R. K. Choudhury, S. S. Kapoor

Nuclaer Physics Division, Bhabha Atomic Research Centre, Mumbai-400085

PhoolChand

Computer Division, Bhabha Atomic Research Centre, Mumbai-400085

(March 14, 1997)

Abstract

A method is suggested for pattern recognition in PHENIX muon arm based on a combined apporach of both artificial neural net (ANN) and principal component analysis (PCA). Best performance is achieved when the data is preprocessed using PCA technique before it is fed to the back propagated neural network.

I. INTRODUCTION

Pattern recognition and track construction is one of the important aspect in high energy heavy ion collisons, particularly when the track multiplicities are quite high. Further, the inhomogenity in magnetic filed and increasing number of tracking detectors make the track finding algorithm more complex as many of the tracks suffer multiple scattering. It is essential to develop some efficient algorithm taking into account the detector positions, the inhomogenity of magnetic field surrounding the detectors , multiple scattering and so on. In this work, we have tried to find out a better track finding technique in an tracking environment where one expect a large track multiplicity. We take an example of the PHENIX detector which is being designed [1,2] for future experiment with Relativistic Heavy Ion Collider at BNL. The main purpose of PHENIX muon arm detector is to detect muon pairs with high mass resolution originating from vector meson decays or virtual photon production. Therefore, a highly efficient pattern recognition technique is desirable as the momentum or the mass resolution of muon pairs will depend on how well the tracks are identified from the set of measured co-ordinates. We generate simulated data for PHENIX muon arm using GEANT based simulation code PISA. Conventionally, methods based on the Principal Component Analysis (PCA) are used for pattern recognition and calssification of true tracks [1,3]. This method works well when the tracking distance is small , number of tracking stations are large and the track multiplicity is not too high. In this work, we show that the PCA method is not good enough in the high multiplicity environment, as many combinations of the co-ordinates not belonging to the true tracks also qualify the PCA test. Next we consider another approach based on Artificial Neural Network (ANN) which has become quite popular in many of the classification problems [4]. However, in this work we found that the ANN based on the normal back propagation algorithm does not work well even it does not converge while learning. To over come the problem of both PCA and ANN, we suggest a new approach based on both PCA and ANN. It is shown that best performance is achieved when the input data is preprocessed using PCA technique before it is fed to the back propagated neural network.

II. SIMULATED TEST DATA

The PHENIX muon arm is designed to give acceptance for particles decaying into lepton pairs emitted at rapidity $y \approx 2$ around the polar angles $10^o \leq \theta \leq 35^o$. The arm has nearly full azimuthal angle coverage. Charged particles entering the muon arm are tracked by three tracking stations, each having three sub-tracking stations. More details are given in ref [1]. The PHENIX detector is modelled using GEANT based packages. We generate the test data for PHENIX muon tracking stations using PISA which is a GEANT based simulation code used for detector simulation. We also used an inbuit J/ψ generator that produces J/ψ which decays into a pair of muons. Fig. 1 shows the momentum distribution just before station I for all those muons which enter the muon arms while corresponding rapidity distribution is shown in fig 2. In the first choice, we take only those muon pairs which penetrate all the three tracking stations. Although, we have chosen muon pairs to start with, the following analysis will also be applicable to pion tracks that penetrate all the three tracking stations. To see how the method works, we have considered this simple case which will be generalised to the more complex situation later on. While passing through the tracking stations a single track will create 27 space points. Since Z position is fixed, we are left with 18, (θ and ϕ) co-ordinates.

III. PRINCIPAL COMPONENT ANALYSIS

The principal component analysis which is also called a "canonical transformation" does a simple co-ordinate transformations to principal axes such that the variances of the new co-ordinates are equal to the eigen values of the co-variance matrix which is generated out of a large sample space [1]. Let us assume that a single track is characterised by a M dimensional hit vector h=$(h_1, h_2......h_M)$ whose average (H_i) and covariance (A_{ij}) are

$$H_i = \frac{1}{N}\sum_{n=1}^{N}(h_i)_n \tag{1}$$

$$A_{ij} = \frac{1}{N}\sum_{n=1}^{N}[(h_i)_n - H_i][(h_j)_n - H_j] \tag{2}$$

where N is the number of hit vectors of the (NXM) dimensinal sample space. Consider the following linear transformation

$$X_j = \sum_{i=1}^{M}\omega_{ij}h_i \tag{3}$$

It can be proved [1] that the variances of X are the eigen values λ of the dispersion matrix A when ω_{ij} represents the eigen vectors of A. Hence

$$Var(X_j) = E[X_j - E(\omega_j h)]^2 = E[\sum \omega_{ij}(h_i - H_i)]^2 = E(\eta_j^2) \tag{4}$$

where E stands for average over N and η_j^2 is given by

$$\eta_j = \sum_{i=1}^{M}\omega_{ij}(h_i - H_i) \tag{5}$$

such that

$$\frac{1}{M}\sum \eta_j^2 = \lambda_j \tag{6}$$

. Therefore, the PCA is tuned with a training data set in order to calculate the avrage and the dispersion matrix H and A respectively. To see if a combination C=$(C_1, C_2......C_{18})$ constitutes a true track, the following generalized distance is calculated

$$\eta_j = \sum_{i=1}^{M}\omega_{ij}(C_i - H_i) \tag{7}$$

$$d = \frac{1}{M}\sum_{j=1}^{M}\frac{\eta_j^2}{\lambda_j} \tag{8}$$

It is now required that for a track candidate $d < d_{max}$. From the previous discussion we know that E$(\eta_j^2) = \lambda_j$ and E(η_j)=0, $\eta_j/\sqrt{\lambda_j}$ will be distributed normally with unit standard deviation. In other words, if we chose $\eta_j/\sqrt{\lambda_j} < 3$ i.e. $d_{max} < 9$, it means 3 standard deviation away from the mean which will occur only in 99.85% of the cases.

IV. PCA RESULTS

As mentioned before, the PHENIX muon arm consists of three tracking stations, each having three sub tracking stations. Therefore, a single track which reaches up to third station will have 9 pairs of θ and ϕ co-ordinates i.e. in total 18 space co-odinates. Since these measurements are in random, one needs to take all possible combinations to calculate the generalised distance 'd' of the hit vactors 'C'. If the track multiplicities at station 1 is N, this results in N^9 combinations. This needs enormous computing time particularly when N is quite large. An alternate approach is to find out the track segments within each station and finally to make a global search to join all the track segments. This procedure will result in $3XN^3$ combinations. The track segment finding proceeds according to the following scheme. First we search for the tracks in a single tracking station assuming various track multiplicities and finally we make a global search to join the track segments. We follow this approach to reduce the computing time . We use a training data set with 116 hit vectors h and a test data set with different hit vectors C i.e. with different multiplicities. Using this training set, the ω_{ij} and λ_j are calculated for each stations as well as for global search. Fig. 3 shows a typical example of the 'd' distribution for station 1 when same data set is used for both training and testing. Although, most probable value of 'd' is around one, still it can be as large as 12. Therefore, if one uses d_{max} around 12 for the test data, all the true tracks will be accepted with 100 percent efficiency. Similar behaviour is also found for station 2 and 3. It is important to note here that if there is any combination of the vector C_i which is not a true track, but its 'd' value is less then d_{max} , it will also add to the true track counting. However, this does not happen within each tracking stations. Table 1 to 3 summarise the results within each station at various d_{max} values. The second column shows the track multiplicities and the third column shows the number of total combinations for which $d \leq d_{max}$. The fourth column shows the number of true tracks out of total tracks for which $d \leq d_{max}$. In all these cases (table 1 to 3), all the tracks for which $d \leq d_{max}$ are true tracks and there is no false contribution for which $d \leq d_{max}$ as it happens for global tracking. It can also be seen that for $d_{max} \approx 12$, the true track recovery efficiency reaches up to 100 percent for all stations.

Next, we join these track segments doing a global searching. Fig. 4 shows the 'd' distribution of the true tracks for global tracking whose value can become as large as 60. In order to have 100 percent track construction efficiency, one must use $d_{max} \leq 60$. However, in case of global tracking, an additonal problem arises due to presence of many false trcaks. There are many false combinations for which $d \leq d_{max}$ and this happens at all d_{max} values although the problem is more serious at higher d_{max} values as well as for higher track multiplicities. Table 4 shows few typical numbers of true and false tracks at multiplicity N=100 for different d_{max} values. Fig. 5 shows the percentage of accepted true and false tracks as a function of multiplicities at a typical d_{max} value of 10 where as fig. 6 shows the plot of the total number of true and false tracks as a function of d_{max} at N=100 (see solid and dashed curves). This shows that at N=100, if one wants 100% true track efficiency, one needs to use a $d_{max} \approx 50$, which will also result in a large number of false tracks. Although this problem is less serious at low multiplicities, still it will result in lots os spurious tracks depending on the d_{max} values.

The contribution from false tracks might get reduced if the number of tracking stations are increased further. To see how the number of sub tracking stations effect the pattern recognition efficiency, we have removed third sub-tracking station from each station. This configuration will result in 12 θ, ϕ co-ordinates for a track that goes all through. Table 5 shows the results. It is surprising to note that whether it is 18 or 12 co-ordinates, the performance does not differ significantly (compare table 4 with that of table 5). In the present case, we have considered only the muon tracks. It will be interseting to see what will happen if we do tracking with pions as it will give additional tracks in between due to deacys. Table 6 shows the performance with only three sub tracking stations, one being at each station. This results in total 6 co-ordinates for a single tracks. In this case the tracking efficiency is very poor as the number of false tracks increase drastically.

V. ARTIFICIAL NEURAL NET (ANN)

The neural network takes a different approach. Consider a three layer network where the output units are denoted by O_i, hidden units by V_j and input units by ξ_k. There are connections ω_{ij} from the inputs to the hidden units and W_{ij} from the hidden units to the output units. Different patterns are labeled by the superscript μ, so input k is set to ξ_k^μ when pattern μ is being presented. Given pattern μ, hidden unit j receives a net input

$$h_j^\mu = \sum_k \omega_{jk} \xi_k^\mu \tag{9}$$

and produces output

$$V_j^\mu = g(h_j^\mu) = g(\sum_k \omega_{jk} \xi_k^\mu) \tag{10}$$

Output unit i thus receives

$$h_i^\mu = \sum_j W_{ij} V_j^\mu = \sum_j W_{ij} g(\sum_k \omega_{jk} \xi_k^\mu) \tag{11}$$

and produces for the final output

$$O_i^\mu = g(h_i^\mu) = g(\sum_j W_{ij} V_j^\mu) = g(\sum_j W_{ij} g(\sum_k \omega_{jk} \xi_k^\mu)) \tag{12}$$

The net output O_i^μ for the pattern μ usually differs from the desired output d_i^μ. In the standard ANN model with backpropagation algorithm [4], one tries to minimise the mean square error defined as

$$E = \frac{1}{2} \sum_{\mu i} [d_i^\mu - O_i^\mu]^2 \tag{13}$$

The aim is to adjust the free parameters of the network ω_{ij} and W_{ij} in such a way that the error function is minimum. Several methods have been proposed to speed up the training of the backpropogation algorithm [5,6]. In this work, we find the ANN based on RPROP algorithm [5] is much faster than the one based on simple backpropagation.

VI. APPLICATION OF ANN FOR PATTERN RECOGNITION

We consider an ANN with 18 input nodes, 18 hidden nodes and 1 output node. The input and hidden nodes are kept same as this combination is found to give better performance. The output is assigned a value either 0 or 1 depending on whether the input vector is a true or a false trcak. For training the ANN, we take 246 hit vectors with nearly equal number of true and false tracks (113+133) so that the training becomes unbiased. We select the input vectors in two different ways. In one case, we take θ and ϕ co-ordinates as input vectors whereas in second case we use PCA transformed co-ordinated d_j as input vectors where d_j is given by

$$d_j = \frac{1}{18} \frac{\eta_j^2}{\lambda_j} \tag{14}$$

such that $\sum d_j = d$.

Therefore, in the first case, the input vector has 18 actual θ, ϕ co-ordinates whereas in second case, the input vector has 18 transformed d_j co-ordinates. The ANN is trained using JETNET-3.4 [7] in batch mode i.e. the connection weights are updated after each epoch of 246 iterations. Fig. 7 shows the plot of error function after each epoch of 246 iterations with both θ, ϕ co-ordinates. It is seen that the learning is very poor with the θ, ϕ choice of inputs and the ANN does not converge even after 50,000 epochs. On the otherhand, with the second choice of inputs, the ANN converges very fast even within 250 epochs. After the training is over, the weights are frozen and the method is applied to the test data. Fig. 8 and fig. 9 show the ANN output distribution after 5000 and 250 epochs with two choices of inputs. Although the desired outputs are 0 and 1, the output values are broadly distributed between 0 and 1 when the θ, ϕ are used as inputs. On the other hand, with d_j as inputs, the learning is not only faster, but also the output produces either 0 or 1 after 250 epochs as shown in fig. 9. Therefore, we use this transformed co-ordinates d_j both for training and testing. We use different sets of test data having different number of true and false tracks. The idea is whether this ANN can differentiate between true and false tracks which are inherently present after PCA test for global tracking (see table 4). Therefore we apply this ANN to the output of table 4. The performance of ANN is summarised in table 7. Here the first column reprsents the total number of tracks that are fed to ANN. The second and third columns show the total number of true and false tracks out of total that are fed to the ANN. The fourth and fifth columns show the number of tracks that are accepted by the ANN for which the output is ≤ 0.5. The sixth and seventh column show the number of true and false tracks that are rejected by the ANN for which the ANN output is > 0.5. The ANN results are also shown in fig. 5 and fig. 6 for comparison (see solid and dashed curves joined with circles). It can be seen that, although the ANN further rejects some of the true tracks, the false track contribution is reduced to almost one or two percent. Here we have not optimised on the neural net performance. Once we optimize, the false track contribution can be reduced to almost 0 while the true track construction efficieny can be increased almost up to 99 percent. More work is in progress.

VII. CONCLUSION

The PCA and the ANN techniques have been applied for muon track recognition. It is found that the PCA method gives best performance when the track multiplicity is low. At higher multiplicities, the false track contibution increases which becomes quite significant if the number of tracking stations are reduced further. In the present analysis, whether it is 18 co-ordinates (9 sub tracking stations) or 12 co-ordinates (6 sub tracking stations), the PCA performance does not differ much. The ANN method has also been applied for pattern recognition. It is found that the ANN can not be trained with actual θ,ϕ co-ordinates of the true and false tracks. However, the ANN learns much faster when the actual θ,ϕ co-ordinates are transformed to a new set of co-ordinates using PCA technique. This combined approach based on both PCA and ANN seems to give better performance.

TABLES

TABLE I. Station no 1:

d_{max}	MULTIPLICITY	TRACKS RECOVERED	TRUE TRACKS	FALSE TRACKS
3.0	100	91	91	0
3.0	75	68	68	0
3.0	50	46	46	0
3.0	25	23	23	0
4.0	100	94	94	0
4.0	75	70	70	0
4.0	50	48	48	0
4.0	25	24	24	0
5.0	100	94	94	0
5.0	75	70	70	0
5.0	50	48	48	0
5.0	25	24	24	0
12.0	100	100	100	0
12.0	75	75	75	0
12.0	50	50	50	0
12.0	25	25	25	0

TABLE II. Station no 2:

d_{max}	MULTIPLICITY	TRACKS RECOVERED	TRUE TRACKS	FALSE TRACKS
3.0	100	96	96	0
3.0	75	72	72	0
3.0	50	48	48	0
3.0	25	24	24	0
4.0	100	96	96	0
4.0	75	72	72	0
4.0	50	48	48	0
4.0	25	24	24	0
5.0	100	97	97	0
5.0	75	73	73	0
5.0	50	49	49	0
5.0	25	24	24	0
6.0	100	98	98	0
6.0	75	74	74	0
6.0	50	50	50	0
6.0	25	25	25	0
7.0	100	100	100	0
7.0	75	75	75	0
7.0	50	50	50	0
7.0	25	25	25	0

TABLE III. Station no 3:

d_{max}	MULTIPLICITY	TRACKS RECOVERED	TRUE TRACKS	FALSE TRACKS
3.0	100	97	97	0
3.0	75	73	73	0
3.0	50	49	49	0
3.0	25	24	24	0
4.0	100	98	98	0
4.0	75	74	74	0
4.0	50	49	49	0
4.0	25	24	24	0
5.0	100	100	100	0
5.0	75	75	75	0
5.0	50	50	50	0
5.0	25	25	25	0

TABLE IV. Global search for three tracking stations with 18 co-ordinates , Multiplicity M=100.

d_{max}	TRUE TRACKS	FALSE TRACKS	TOTAL
2.0	86	4	90
3.0	88	12	100
4.0	91	19	110
5.0	94	20	114
6.0	94	21	115
8.0	95	39	134
10.0	96	60	156
15.0	96	100	196
20.0	97	133	230
30.0	98	228	326
50.0	99	393	492

TABLE V. Global search for three tracking stations with 12 co-ordinates , Multiplicity M=100.

d_{max}	TRUE TRACKS	FALSE TRACKS	TOTAL
2.0	87	6	93
3.0	91	15	106
4.0	92	20	112
6.0	95	28	123
10.0	95	64	159
15.0	96	100	196
20.0	97	137	234
30.0	99	227	326
50.0	100	395	495

TABLE VI. Global search for three tracking stations with 6 co-ordinates , Multiplicity M=100.

d_{max}	TRUE TRACKS	FALSE TRACKS	TOTAL
2.0	93	1049	1142
4.0	97	2441	2538
6.0	99	3926	4025
8.0	100	5381	5481

TABLE VII. Performance of ANN

TOTAL TRACKS	TRUE TRACKS	FALSE TRACKS	TRUE ACCEPTED	FALSE ACCEPTED	TRUE REJECTED	FALSE REJECTED
492	99	393	92	4	7	389
326	98	228	92	4	6	224
230	97	133	92	3	5	130
196	96	100	92	3	4	97
156	96	60	92	3	4	57
134	95	39	92	2	3	37
115	94	21	92	2	2	19
110	91	19	89	2	2	17

REFERENCES

[1] PHENIX Conceptual Design Report, 1993, Unpublised.

[2] PHENIX/SPIN Conceptual Design Report, 1995, Unpublised.

[3] M. G. Kendall and A. Stuart, " The advanced theory of statistics", third edition (Ch. Griffin and company, Ltd, London, 1976) V3 P294.

[4] J. Hertz, A. Krogh and R. G. Palmer, "Introduction to the theory of neural computation", Lecture notes Vol. I, Addison-Wesley publishing co. 1991.

[5] Martin Riedmiller and Heinrich Braun, A Direct Adoptive Method for faster Backpropagation Learning : The RPROP Algorithm, Proceedings of the IEEE International conference on Neural Networks (ICNN), p286 - 591, San Fransisco, CA, March, 28 - April 1, 1993.

[6] J. Reifman, J. Vitela and J. C. Lee, Statistical and Optimisation methods to Expedite neural network Training for Transient Identification , Topical Meeting on Nuclear Plant Instrumentation, Control and Man-machine Interface Technologies, April 18-21, 1993, Oak Ridge, Tennessee.

[7] Carsten Peterson, T. Rongvaldsson and Leif Lonnblad, "JETNET 3.0- A versatile Artificial Neural Network Package", LU TP 93-29, CERN-TH 7135/94.

FIGURES

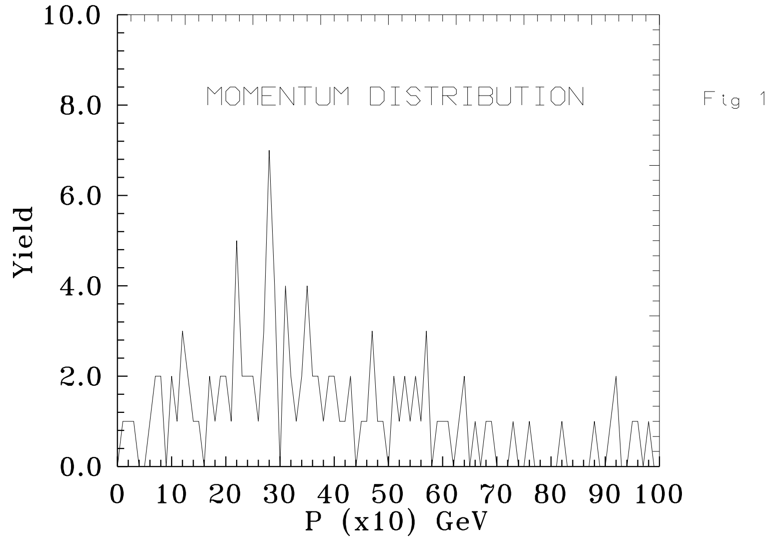

FIG. 1. Momentum distribution of Muons before station I

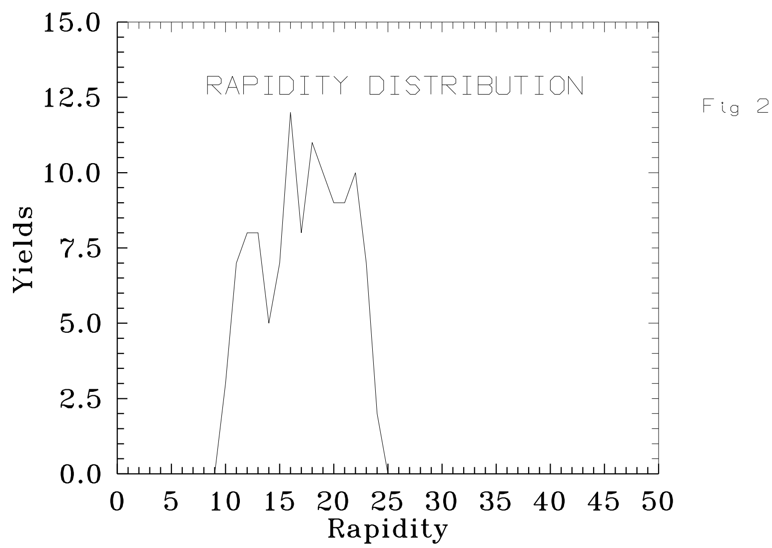

FIG. 2. Rapidity distribution of Muons that enter the PHENIX Muon arm

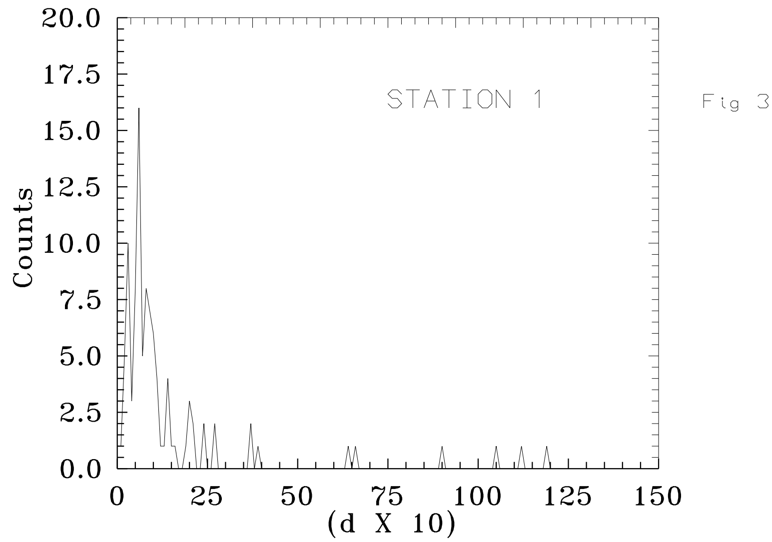

FIG. 3. The normalised distance 'd' distribution for tracking station 1. The 'd' values are multiplied by a factor of 10.

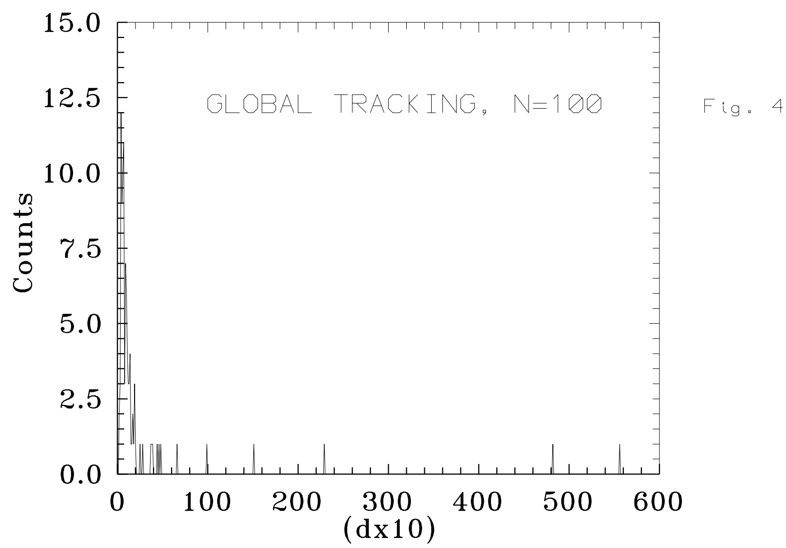

FIG. 4. The normalised distance 'd' distribution for global tracking. The 'd' values are multiplied by a factor of 10.

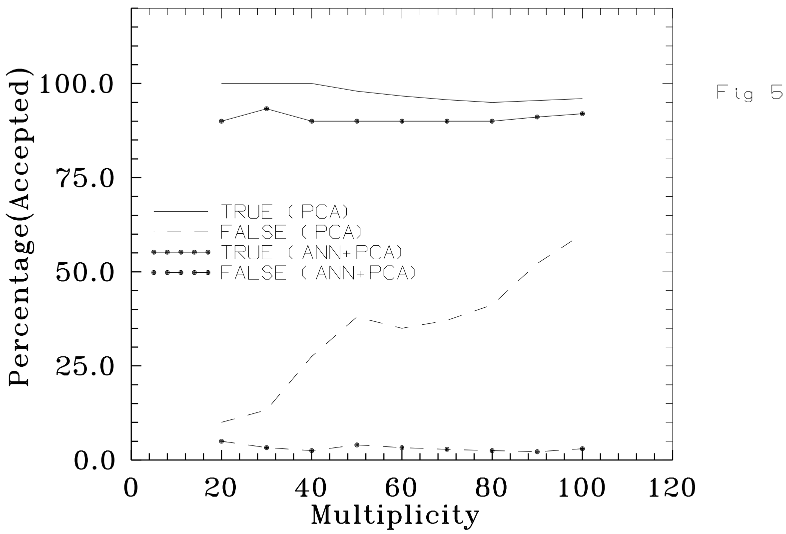

FIG. 5. The percentage of accepted true and false tracks at N=100 as a function of d_{max} for global tracking both for PCA and ANN.

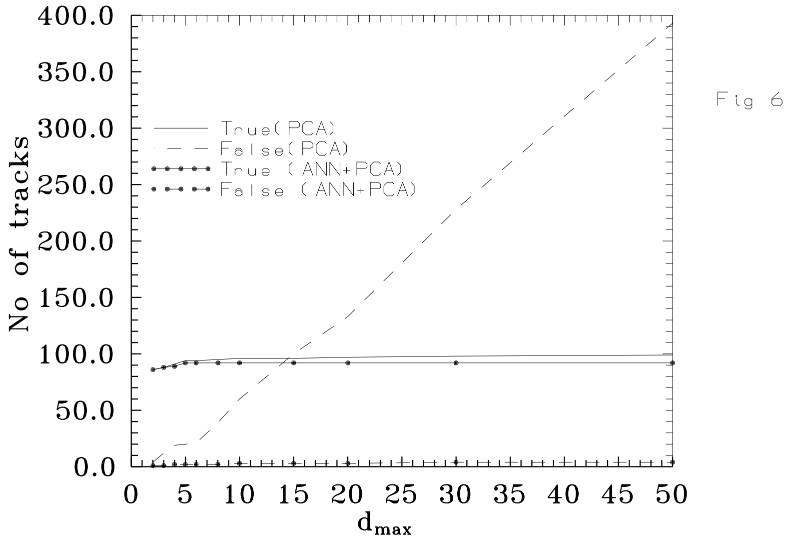

FIG. 6. The total number of true and false tracks as a function of track multiplicity both for PCA and ANN. The typical $d_{max} \approx 10$.

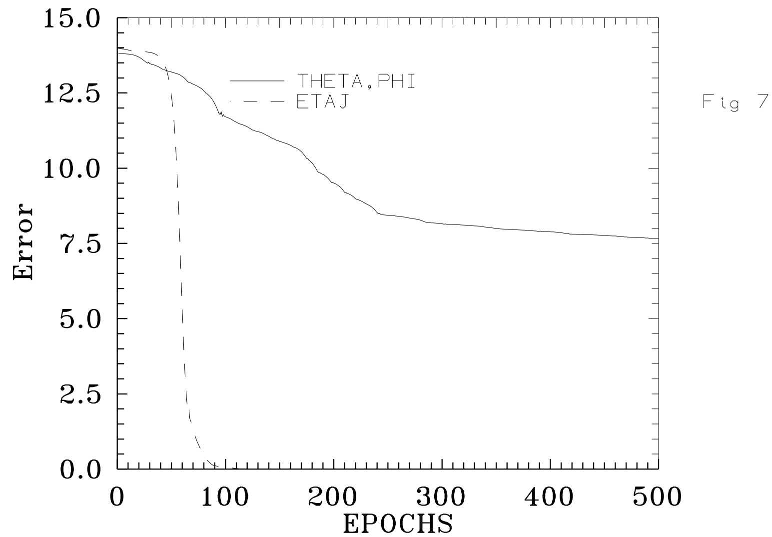

FIG. 7. The Error function versus epoch numbers for two types of inputs

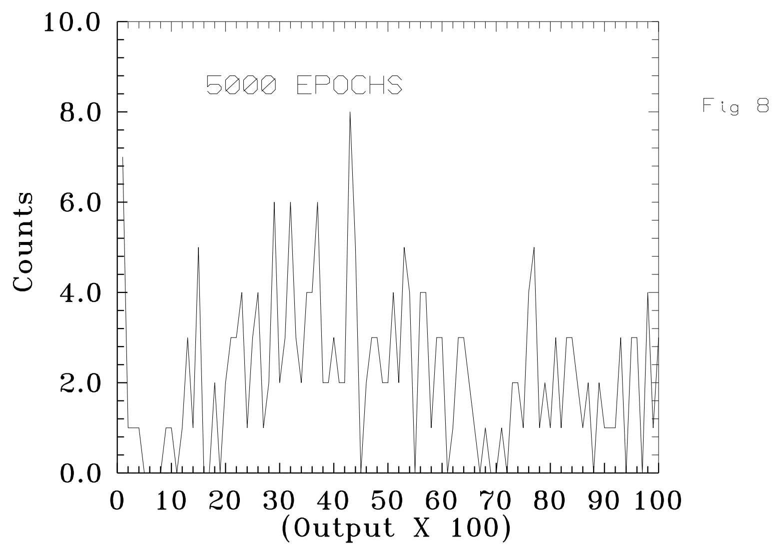

FIG. 8. The ANN output distribution after 5000 epochs when the actual θ,ϕ track co-ordinates are fed as net inputs. The X-axis is multiplied by a factor of 100.

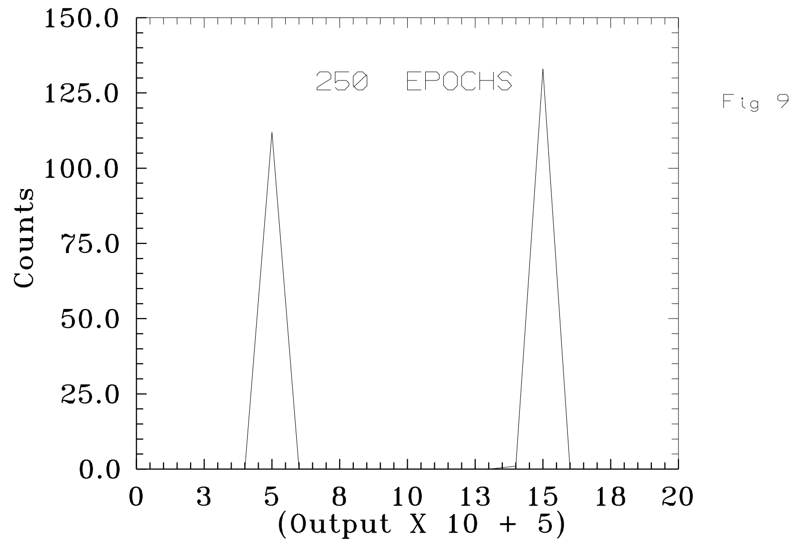

FIG. 9. The ANN output distribution after 250 epochs when PCA transformed co-ordinates are fed as net inputs. The X-axis is multiplied by a factor of 10 and shifted by 5 units towards right.