AN ADAPTIVE DATA SORTER BASED ON PROBABILISTIC NEURAL NETWORKS

C. David Wang and James P. Thompson

AIL Systems Inc.
Melville, NY 11747

ABSTRACT

Based on a self-organized, probabilistic neural network (PNN) paradigm, a parallel network can be used to sort data parameters into classes with high-sorting accuracy and low fragmentation. The capabilities of the sorter, as applied to ESM pulse-data sorting, will be shown.

The PNN implements the statistical Bayesian strategy by computing a joint probability density over all input data parameters to match a group of candidate data classes. The sorting is accomplished by assigning the inputs to the most likely group with highest probability density estimate. Based on test data from an ESM system, the PNN has shown significant improvement over conventional rule-based techniques.

The parallel computer architecture of PNN is well-suited for very large-scale integration (VLSI) chip implementation. An 80,000 gate semicustom chip design concept will be described.

INTRODUCTION

Modern electronic support measure (ESM) systems must handle high-pulse arrival rates and deinterleave pulse trains in a dense environment. Very often, the system must operate with corrupted data, occasional pulse dropout, and unknown PRI modulations.

Conventional pulse sorting techniques can be described as sequential rule checking systems. Pulse parameters such as frequency, amplitude, pulse width, and angle of arrival (AOA) are compared against previously established pulse groups called emitter bins. Each parameter is checked serially to see whether it falls within some tolerance of an existing bin. When all rules are satisfied, the pulse is assigned to the bin; otherwise, a new bin will be created for the pulse. The next step in pulse sorting is to upgrade the tolerance levels based on current statistics such as mean and standard deviation measures.

The rule-based systems have encountered several fundamental problems. Firstly, they are slow because rules are applied sequentially and because a parallel rule-checking algorithm is difficult to implement. Secondly, they do not provide a quality measure or how closely the input parameters are matched to these bins. Thirdly, they respond rather poorly to anomaly situations such as noisy, incomplete data and corrupted pulse parameters. Lastly, they cannot tolerate simple equipment failures. Very often, a small defect in hardware can cause the system to fail.

An adaptive neural network (ANN) is a parallel and distributed computer which is composed of many simple processing elements (PE's) to achieve a common goal. The PE's are designed to mimic a group of biological neurons which can learn from training patterns and respond to practical and noisy scenarios in an adaptive manner. Unlike the sequential rule-checking approach, the neural network makes its decision based on all parameters collected in the past. If one of the incoming pulse parameters is corrupted due to dropout or measurement noise, the pattern-recognition capability is not seriously impaired.

Instead of programming the neural network in the conventional manner, the neural network are "taught" to accept a set of input patterns and produce acceptable answers. The network parameters are adjusted over and over until the all outputs are satisfied. When they are properly "trained," ANN's can provide several desirable benefits:

- Achieve real-time system response
- Produce approximate results from noisy, incomplete data
- Provide bin matching measures such as probabilities
- Accomplish limited fault-tolerant computations

The PNN that estimates the probability density functions from a set of training patterns was presented by D. Specht (references 1 – 3). The PNN has been successfully applied to hull-to-emitter correlation problems for electronic intelligence systems. They have also reported two orders of magnitude of savings in training

CH3007-2/91/0000-1096 $1.00 © 1991 IEEE

time against the popular back propagation network, while maintaining comparable pattern recognition accuracy.

As shown in Figure 1, the PNN consists of an input layer, a hidden layer, and an output layer. Each layer contains a number of simple processing elements or artificial neurons. Training of the PNN is fast because the parameters used in the hidden layer are simply normalized versions of the training data. The PNN essentially accomplishes its training in one step. Therefore, it is ideal for real-time pulse sorting applications.

Based on the PNN model, a neural network for sorting radar pulses was developed by the authors of this paper in 1989. The key issue was that a priori training data sets such as the AOA parameter for the network would not be available. The system must be self-organized to obtain its own coefficients for the hidden layer by means of internal competitions among groups of processing elements. The "winner" of the competition is the one which has the largest probability. After the first set of patterns is obtained, it can be refined as more data becomes available.

Self-Organized Probabilistic Neural Network Concept

The PNN was selected because it approaches the Bayesian optimal solution, and it can be self-organized in real time without a priori knowledge of the environment. The PNN implements the statistical Bayesian strategy which computes probability density functions (PDF's) of a group of pulse classes. The sorting is accomplished by associating the input to the pulse group with the highest probability.

The simplified block diagram of the pulse sorter is shown in Figure 2. The system is divided into several identical subunits called cluster processors. Based on a single set set of similar pulse parameters presented to the neural network, the cluster processor computes a PDF for that group. The theoretical basis for the PNN is the Bayesian strategy for decision rules used to classify patterns. The mathematical expression for PDF estimation (1) for an emitter class is shown below:

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} \exp\left\{ -\frac{[x - w(i)]' \; [x - w(i)]}{2\sigma^2} \right\} \qquad (1)$$

where $f(x)$ = probability density function for input x
x = pulse parameter vector
m = total number of training patterns
$w(i)$ = the ith training weight vector
i = index for the training pattern vector $w(i)$
σ = smoothing parameter $(0 < \sigma < 1)$
x' = transposition of a vector x

The input vector x is a collection of measured pulse data parameters expressed by:

x = [FF, PA, PW, AOA]
where FF = fine frequency
PA = pulse amplitude
PW = pulse width
AOA = angle of arrival

For certain ESM systems, the AOA measurements are expressed by relative phases between three antenna elements to a reference antenna; that is, the AOA is itself a vector of three components:

AOA = [coarse AOA, fine AOA, extra-fine AOA]
= [CRS, FIN, XFN]

The AOA's are the most reliable sorting parameters because other parameters can be varied from pulse to pulse. During the measurement dwell interval, the variation of AOA is usually small. Fine frequency (FF) and amplitude are the next most important parameters for deinterleaving and pulse width is the least reliable parameter because the data can be corrupted due to multipath transmissions. Multipath effects can distort the pulse envelope by creating a tail to the pulse.

A Simple Example of Two-Class PDF Estimation

The following is a simple example of estimating the PDF's associated with two pulse groups. Assume the inputs are measured center frequencies from two emitter classes x_1 and x_2; that is,

$x = [x_1, x_2]$
where $x_1 = [2\ 3\ 4]$
$x_2 = [5\ 6\ 7\ 8\ 9]$

Note that the samples are uniformly distributed over two different frequency ranges. Thus, one expects that the estimator must produce two PDF's with uniform probability distributions. From the distributions, we can measure bin statistics such as mean, standard deviation, probability of detection (PD), and probability of false alarm (PFA).

Figure 3 is a family of PDF estimates as a function of the smoothing function σ ranging from 0.2 to 0.8. As seen from these figures, a σ of 0.4 is seen to produce the largest PD and the smallest PFA. In practice, σ for each emitter group is initially set to 0.4, and it is refined as more pulses become available.

Field Test Data Verification

The pulse sorting algorithm has been successfully applied to a group of measured data from an existing ESM system. Table 1 is an example of the input data and corresponding output probability estimates. The seven columns on the left-hand side are input pulse parameters as measured from the acquisition receiver. As

Table 1. A Pulse Sorting Example

N	FF	PW	PA	CRS	FIN	XFN	A	B	PDF Estimates			
1	75	45	60	41	117	105	1	1	100			
2	105	45	90	112	101	101	2	2	10	0		
3	60	45	60	29	107	101	1	1	85	4		
4	105	45	90	109	101	98	2	2	8	99		
5	0	30	45	103	26	8	3	3	0	0	0	
6	75	45	60	32	115	113	1	1	91	7	0	
7	105	45	90	110	103	98	2	2	8	99	0	
8	75	60	60	38	113	106	1	1	88	9	0	
9	0	15	45	98	9	11	3	3	0	0	84	
10	105	45	90	115	108	103	2	2	7	97	0	
11	60	45	60	51	125	111	1	1	80	10	0	
12	105	45	90	110	105	100	2	2	9	99	0	
13	0	15	45	110	15	5	3	3	0	0	90	
14	75	45	60	48	125	113	1	1	86	13	0	
15	105	45	90	112	105	100	2	2	9	99	0	
16	0	15	45	5	17	73	4	0	0	0	1	0
17	75	45	60	35	117	104	1	1	91	8	0	0
18	105	45	90	114	106	100	2	2	8	99	0	0
19	75	45	60	43	110	105	1	1	91	12	0	0
20	0	30	45	109	23	20	3	3	0	0	87	1
•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•
41	105	45	75	110	107	101	2	2	13	92	0	0
42	105	45	75	114	104	98	2	2	10	92	0	0
43	75	45	45	42	110	97	1	1	84	8	0	0
44	105	45	75	108	102	100	2	2	13	92	0	0
45	75	45	45	30	115	94	1	1	84	5	0	0

seen from the table, the pulse descriptor is a 33-bit word consisting of six parameters:

Parameter Description		Range	No. of Bits
FF	Fine Frequency	0 – 7	3
PW	Pulse Width	0 – 15	4
PA	Pulse Amplitude	0 – 31	5
CRS	Coarse AOA Phase	0 – 127	7
FIN	Fine AOA Phase	0 – 127	7
XFN	Extra Fine AOA Phase	0 – 127	7
		Total:	33

The two columns labeled A and B are pulse sorter outputs from the existing rule-based ESM system and the proposed PNN technique, respectively. The last four columns are probability density estimates for four pulse clusters. [Note that the bin selections were based on the maximum probability estimates. However, when all probabilities were below a threshold (e.g., less than 20 percent as shown in pulse Nos. 2, 5, and 16), a new bin was created for each case.]

Judging from the two decision columns (labeled A and B), we see that the two algorithms essentially achieve the same sorting accuracy. The advantage of the PNN approach is a quantitative measure of the probability function associated with each pulse descriptor. The result of the pulse sorting process is tabulated in a final bin report as shown in Table 2.

Table 2. Final Bin Report

Data Class 1						
1	5	3	4	41	117	105
2	4	3	4	29	107	101
3	5	3	4	32	115	113
4	5	4	4	38	113	106
5	4	3	4	51	125	111
6	5	3	4	48	125	113
7	5	3	4	35	117	104
8	5	3	4	43	110	105
9	5	3	4	29	104	99
10	5	4	4	40	115	105
11	6	3	4	40	119	102
12	4	3	4	35	109	98
13	4	3	4	36	111	106
14	5	3	4	21	97	93
15	5	3	3	40	113	110
16	5	3	4	50	117	105
17	5	3	3	42	110	97
18	5	3	3	30	115	94
	4.83	3.11	3.83	37.78	113.28	103.72
	0.51	0.32	0.38	7.85	6.83	5.95*
Data Class 2						
1	7	3	6	112	101	101
2	7	3	6	109	101	98
3	7	3	6	110	103	98
4	7	3	6	115	108	103
5	7	3	6	110	105	100
6	7	3	6	112	105	100
7	7	3	6	114	106	100
8	7	3	6	113	107	101
9	7	3	6	111	104	101
10	7	3	6	114	103	98
11	7	2	6	113	106	102
12	7	3	6	115	109	101
13	7	3	6	111	107	101
14	7	3	6	111	106	100

*Mean Standard Deviation

Table 2. Final Bin Report (Cont)

Data Class 2 (Cont)						
15	7	3	6	117	106	101
16	7	3	5	110	107	101
17	7	3	5	114	104	98
18	7	3	5	108	102	100
	7.00	2.94	5.83	112.17	105.00	100.22
	0.00	0.24	0.38	2.38	2.33	1.44
Data Class 3						
1	0	2	3	103	26	8
2	0	1	3	98	9	11
3	0	1	3	110	15	5
4	0	2	3	109	23	20
5	0	1	3	106	12	20
6	0	2	3	110	15	4
7	0	1	3	94	23	33
8	0	1	3	97	13	7
	0.00	1.38	3.00	103.38	17.00	13.50
	0.00	0.52	0.00	6.37	6.16	10.04
Data Class 4						
1	0	1	3	5	17	73
	0.00	1.00	3.00	5.00	17.00	73.00
	0.00	0.00	0.00	0.00	0.00	0.00

Summary of PNN Sorting Performance

The following table is a summary of the sorting performance of the PNN compared against a rule-based sorter (RBS). The data base for the study consists of over 3,000 pulse descriptor words (PDW's) from an existing ESM system data file.

Parameters	IRAD Performance (PNN)	Current Technology (RBS)
No. of Pulses Sorted Over Total	3,057/3,172	2,978/3,172
Fragmentation Index (No. of Bins)	261	378
Sorting Accuracy	96.3%	93.8%

Analysis of the two methods reveals that the RBS method creates 40 percent more bins that the PNN. In terms of sorting accuracy, the PNN method is 2.5 percent better than the RBS. Details will be provided in the following paragraphs.

Based on over 3,000 pulse data, the PNN has achieved more accurate sorting performance than conventional rule-based sorting techniques. Potential improvements in processing speed, size, weight, and power dissipation will be evaluated based on an application specific integrated circuit (ASIC) design. A VLSI chip composed of over 80,000 gates is viewed as feasible for future hardware implementation.

Pulse-to-Pulse PDF Estimation and Display

Since the PNN measures the PDF for each pulse group, the PDF's can be drawn graphically on a pulse-by-pulse basis. This is shown in a sequence of 3-D PDF plots and contour maps for the following five measured pulses:

Pulse Index	Fine Freq	Pulse Width	Pulse Ampl	Coarse	AOA Fine	X-Fine	Pulse Group
1	5	3	4	41	–11	–23	1
2	7	3	6	–16	–27	–27	2
3	4	3	4	29	–21	–27	1
4	7	3	6	–19	–27	–30	2
5	0	2	3	–25	26	8	3

Figures 4 through 8 display the PDF estimates in a cumulative fashion. Since it is impossible to draw a 6-D graph, the PDF's are shown by multiple two-variable graphs. For example, Figure 4 shows a 3-D plot for coarse vs. fine AOA. From these figures, one can measure statistical properties between these clusters and determine the sorting performance in a quantitative manner.

Specification of a VLSI Neurochip for Sorting

A preliminary specification for a pulse sorting VLSI neurochip is listed below. The chip contains three emitter clusters and is cascadable for more clusters. Total gate count for the device is approximately 80,000 gates. The parallel distributed processing architecture is feasible for low I/O pad (less than 100 pins) digital VLSI implementation. It is also suitable for high throughput (over 20 MPPS) ESM applications.

Specification of the Pulse Sorting Neurochip	
Feature Size	1.0 μm
Logic Family	CMOS
Clock Rate	20 MHz
Estimated Chip Density	80,000 gates
No. of Signal I/O Pins	100
Estimated Power Dissipation	3 W
Peak Pulse Arrival Rate	20 MPPS
No. of Emitter Bins Per Chip	3

REFERENCES

1. Specht, D.F., "Probabilistic Neural Networks," Neural Networks, Vol. 3, 1990, pp. 109 – 18.
2. Specht, D.F., "Probalistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification," IEEE Trans on Neural Networks, Vol. 1, No. 1, March 1990, pp. 111 – 21.
3. Maloney, P. S., "An Application of Neural network Technology to Surveillance Information Correlation and Battle Outcome Prediction," 1989 NAECON Digest, May 1989, pp. 948 – 55.

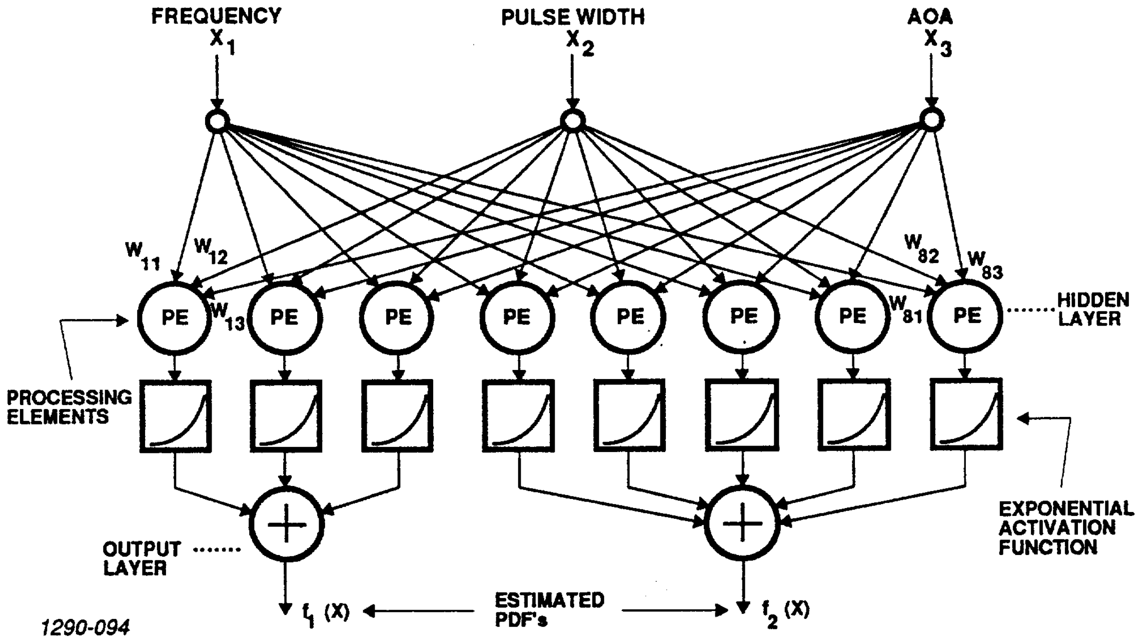

Figure 1. Block Diagram of Probabilistic Neural Network

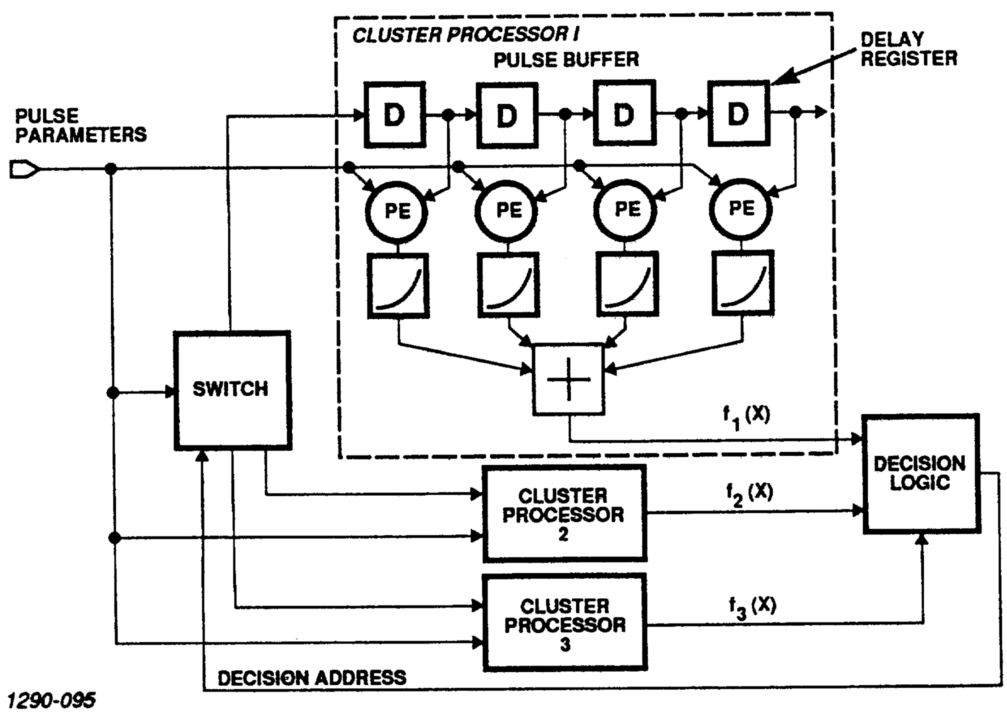

Figure 2. Pulse Sorting PNN Architecture

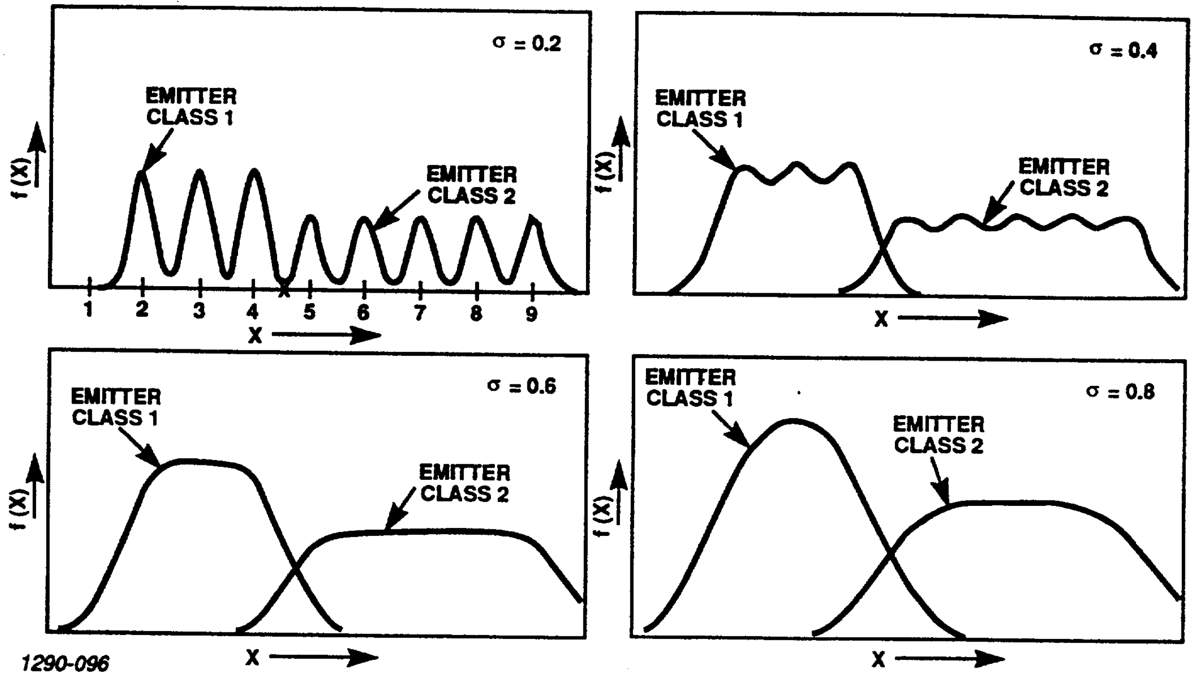

Figure 3. Probability Density Function Estimation as a Function of Smoothing Parameter σ

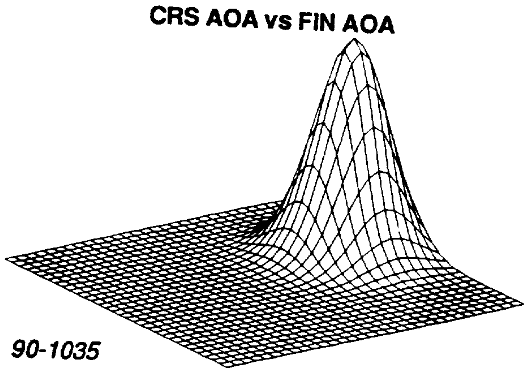

Figure 4. PDF Estimates for Pulse 1

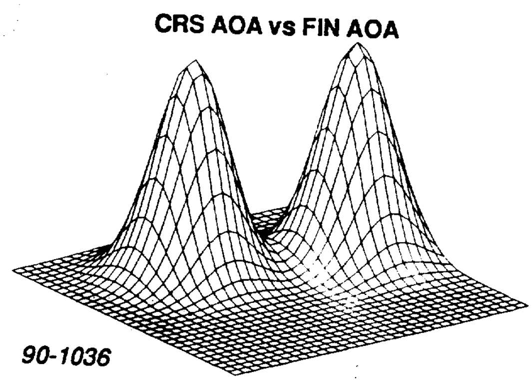

Figure 5. PDF Estimates for Pulse 2

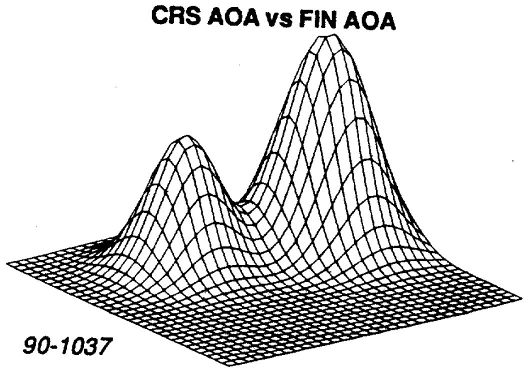

Figure 6. PDF Estimates for Pulse 3

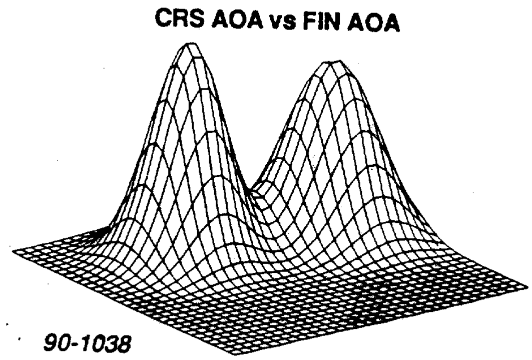

Figure 7. PDF Estimates for Pulse 4

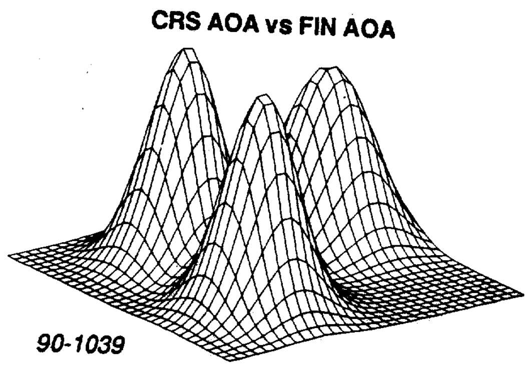

Figure 8. PDF Estimates for Pulse 5