# Artificial Neural Network Analysis of Polygraph Signals

John E. Angus, Ph.D. and Patrick F. Castelaz, Ph.D.
Claremont Graduate School
Claremont, CA 91711-3988

October 1993

Department of Defense Polygraph Institute
Fort McClellan, Alabama 36205-5114
Telephone: 205-848-3803
FAX: 205-848-5332

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



**1. AGENCY USE ONLY** *(Leave blank)*

**2. REPORT DATE**
October 1993

**3. REPORT TYPE AND DATES COVERED**
Final Report (From Jul 92 to Oct 93)

**4. TITLE AND SUBTITLE**
Artificial Neural Network of Polygraph Signals

**5. FUNDING NUMBERS**
DoDPI93-R-0010

**6. AUTHOR(S)**
John E. Angus and Patrick F. Castelaz

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
The Claremont Graduate School
Department of Mathematics
Blaisdell House, 143 E. Tenth St.
Claremont, California 91711-3988

**8. PERFORMING ORGANIZATION REPORT NUMBER**
N00014-93-C-0171

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Department of Defense
Polygraph Institute
Building 3165
Fort McClellan, AL 36205-5114

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
DoDPI93-R-0010

DoDPI92-P-0008

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Public release, distribution unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

The purpose of this research was to investigate the use of artificial neural networks (ANN) in classifying psychophysiological detection of deception (PDD) examinations as deceptive or non-deceptive. ANNs are mathematical models of the computing architecture of the human brain. An ANN was designed to accept all four signals (galvanic skin resistance, cardiovascular activity, thoracic respiration and abdominal respiration) from the polygraph output in their entirety. The PDD data used in the study consisted of confirmed Zone Comparison Technique (ZCT) examinations of 56 subjects, of which only 15 were non-deceptive. The ANN application resulted in an 87% correct classification of non-deceptive subjects and a 95% correct classification of deceptive subjects. The misclassifications were evenly split: 2 misclassified deceptives (out of 41) and 2 misclassified non-deceptives (out of 15). The two non-deceptives were just slightly over the classification threshold, into the deceptive region of the classification space, and could potentially be called inconclusive. While these results are promising, they are based on a limited set of data, so generalization to a claim that they will successfully address the overall polygraph classification problem requires more extensive evalation and demonstration on a much larger database of subjects.

**14. SUBJECT TERMS**
artificial neural networks, polygraph, signal procession, algorithms, psychophysiological detection of deception

**15. NUMBER OF PAGES**
77

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | |

DoDPI93-R-0010

Final Report
CDRL 0001AD
Contract No. N00014-93-C-0171

# Artificial Neural Network Analysis Of Polygraph Signals

Prepared by
Dr. John E. Angus
Dr. Patrick F. Castelaz

for
The Office of Naval Research

Mr. Howard Timm
Scientific Officer
PERSEREC

October 30,1993

## Director's Foreword

One of the key means of improving the accuracy of the psychophysiological detection of deception (PDD) techniques is computer analysis of PDD data. Computers can analyze factors that are impossible for even the most competent of human examiners to see, no matter how thoroughly he or she inspects the data. Computers can analyze complex waveforms far faster, in much greater detail, and far more consistently than humans.

It is no easy task to determine the best way to analyze the test data. Many statistical approaches have been used, with varying success. The first major approach taken was discriminant analysis to differentiate between innocent and guilty subjects. Other avenues being explored include decision trees, logistic regressions, and fuzzy logic. If we are to find the best approach, we must explore all avenues.

The approach taken in this study is artificial neural network (ANN) analysis. ANNs are a mathematical attempt to mimic the functioning of the human brain, which uses biological neural networks. The conventional computer processes information serially. That is, one operation is conducted after another, sequentially, and each operation is completed before the next is started. On the other hand, the brain processes information in parallel; many operations are going on simultaneously, and the progress of one operation can affect the progress of others. Artificial neural networks also processes in parallel, and are thus able to "learn" how to analyze charts without identifying the criteria for evaluation.

This ability to learn on their own without explicit instructions about what to look for opens the possibility of having computers find novel indices of deception in PDD data. Clearly, this avenue must be investigated if we are to improve the accuracy of PDD decisions.

The procedures used in this study correctly identified 95% of the deceptive subjects and 87% of the truthful subjects. The authors believe this represents the lower bounds on the potential performance of ANNs, as they were limited by a very small amount of data from truthful subjects. The small number of subjects is an important factor limiting the generalizability of the results of this study.

Michael H Capps

Michael H. Capps
Director

Abstract

ANGUS, J.E., and CASTELAZ, P.F. Artificial neural network analysis of polygraph signals. October 1993, Report No. DoDPI93-R-0010. Department of Defense Polygraph Institute, Ft. McClellan, AL 36205. The purpose of this research was to investigate the use of artificial neural networks (ANN) in classifying psychophysiological detection of deception (PDD) examinations as deceptive or non-deceptive. ANNs are mathematical models of the computing architecture of the human brain. An ANN was designed to accept all four signals (galvanic skin resistance, cardiovascular activity, thoracic respiration and abdominal respiration) from the polygraph output in their entirety. The PDD data used in the study consisted of confirmed Zone Comparison Technique (ZCT) examinations of 56 subjects, of which only 15 were non-deceptive. The ANN application resulted in an 87% correct classification of non-deceptive subjects and a 95% correct classification of deceptive subjects. The misclassifications were evenly split: 2 misclassified deceptives (out of 41) and 2 misclassified non-deceptives (out of 15). The two non-deceptives were just slightly over the classification threshold, into the deceptive region of the classification space, and could potentially be called inconclusive. While these results are promising, they are based on a limited set of data, so generalization to a claim that they will successfully address the overall polygraph classification problem requires more extensive evaluation and demonstration on a much larger database of subjects.

Key-words: artificial neural networks, polygraph, signal procession, algorithms, psychophysiological detection of deception

# Artificial Neural Network Analysis of Polygraph Signals

**Final Report - CDRL Item No. 0001AD**

**Contract No. N00014-93-C-0171**

**30 October 1993**

**Dr. John E. Angus**

**Dr. Patrick F. Castelaz**

## Introduction and Background

This report describes research undertaken during the period 29 June 1993 through 30 October 1993 under ONR contract N00014-93-C-0171, performed in response to the Broad Agency Announcement in the area of Forensic Psychophysiology: Detection of Deception, dated 9 July 1992.

The purpose of this research is to investigate the use of Artificial Neural Networks in classifying polygraph charts (examinations) as deceptive or non deceptive.

Recently, the National Security Agency and Department of Defense Polygraph Institute have shown increased interest in the use of quantitative methods to assist examiners in scoring polygraph charts. This interest is partly due to the need for standardization of the scoring process, the need for increased accuracy of scoring, and the desire to decrease the number of inconclusive test results. Concurrently, computerized polygraph workstations have been developed that collect, display, and store raw polygraph signal data in real time. Two such systems are the CAPS (Computer Assisted Polygraph System, Raskin et.al. 1988; Kircher et.al. 1988) and the PC based commercial system developed by Axciton Systems. Both systems provide scoring algorithms, and have greatly facilitated the on going research interest in quantitative scoring of polygraphs.

Landmark research by Olsen, Ansley, Feldberg, Harris, and Cristion (1991) has demonstrated the efficacy of quantitative methods in this area. They employed the classical logistic regression model to mock crime polygraph data and showed convincingly that the use of quantitative scoring could substantially reduce the percent of inconclusive test results while retaining accuracy that rivals that of trained examiners. Dr. Olsen and his colleagues have conducted more recent research on actual polygraph test results, but results of this research are not available at the time of this report.

The technique employed by Olsen et.al. (1991), logistic regression, is a powerful and flexible technique for modeling the probability of deception as a function of explanatory variables. However, the relevant explanatory variables are not explicitly available, and must be extracted as "features" from the raw polygraph signals in order to apply logistic regression. It is the process of identifying and accurately extracting these relevant features that predominantly determines the success of the logistic regression technique.

Four signals are monitored during a polygraph examination: galvanic skin response (GSR), heart rate / blood pressure (Cardio), thoracic respiration, and abdominal respiration. The Axciton system samples, displays, and records these signals every 1/30 second, displaying a more or less continuous signal for each response. A typical response, beginning with a question and ending with the beginning of the next question lasts roughly 25 seconds, for a total of 750 sample points for 4 signals, a total of 3,000 data points for a single question / response. A trained examiner uses only a fraction of this data, as much of it corresponds to relief as opposed to reaction. Perhaps 5 to 7 seconds of data following the question are actually used in scoring.

Examiners are trained to recognize features in these signals that are highly correlated with deception: changes in amplitude, duration, baseline changes, narrowing of signals, and so on. Olsen et.al. (1991) quantified and extracted a large number of features automatically, and used these as independent variables in the logistic regression model. Of course, these features were not arbitrarily chosen, but based on studies, scientific precedent, and expertise supplied by trained examiners. Even so, the logistic regression model must, through stepwise fitting techniques, "learn" how to assign importance and weight to the four signals and their features from a training database. The result of this process was, as mentioned previously, very impressive. However, the question arises as to whether the feature extraction process, and / or the structure of the logistic regression model itself, can be improved to yield even better accuracy and lower percentages of inconclusive results.

Artificial Neural Networks (ANNs) are mathematical models of the computing architecture of the human brain. They have the capability to approximate a broader class of surfaces than the logistic regression model, and in fact the logistic regression model can be viewed as a special case of an ANN having an input layer (representing the independent variable inputs), a single middle layer neuron that performs the logistic function, and a single output that represents the result of the logistic transformation. Adding more logistic function processing neurons to the middle layer, and then combining their outputs using one final logistic function, generalizes the standard logistic regression model and adds greater flexibility in the relationships that can be approximated. If the logistic regression model is to some degree deficient in representing the surface that represents the relationship between the features and the probability of deception, then the ANN will improve the accuracy of scoring polygraphs. If the logistic regression model is adequate in this respect, the ANN will do no worse.

It was an hypothesis of this research that the critical factor in improved scoring would be the feature extraction aspect. An ANN can be designed to accept all four signals from the polygraph output in their entirety, and the interconnection weights and number of middle layer neurons adjusted by the training process to represent the features that are necessary for accurate classification. The advantages of this approach seem clear, assuming that the training database is very diverse and of high quality. First, the subjectivity of feature extraction is removed. Second, it becomes possible that the ANN can recognize and make use of features that are overlooked by even trained examiners.

Further discussion of the ANN approach and special considerations associated with it are addressed later in this report.

The success of the ANN approach, and indeed of any quantitatively based approach based on "learning," is dependent on the availability of a quality database. This database must contain a large and diverse class of confirmed polygraph examinations, i.e. examinations for which ground truth has been established (e.g. through confession). Data of this type were provided for this investigation in the form of compressed raw data files from the Axciton workstation. The processing of this data occupied the majority of this effort, as no standard software nor documentation for the Axciton data formats were made available.

The remainder of this report is organized into two major sections: I. Polygraph Data Processing and II. Artificial Neural Network Processing of Polygraph Signals. Section III presents overall summations and conclusions. In describing the data processing, we have tried to document the software developed in this effort so that future researchers will benefit. Thus, source listings and descriptions of the polygraph data made available to us are included. Despite the heavy data processing burden in this study, we have developed a prototype Cellular Automaton (a special case of an ANN) that scores polygraph examinations, and have achieved what we believe are encouraging results that compare well with other scoring (quantitative and otherwise) methods. Results and a Summary of this work are given at the end of section II of this report.

# I. Polygraph Data Processing

### *I.1 Description of Initial Data Provided*

Dr. Dale Olsen of the Johns Hopkins Applied Physics Laboratory, as authorized by the NSA and DODPI, supplied a 90mB data cartridge (readable on a PC equipped with an Iomega Bernoulli hard drive) containing two files compressed using the commercial PC software package known as PKZIP (a product of PKWare, Inc.). These two files, CDE.ZIP and ZCT.ZIP, contain raw polygraph data as recorded by the Axciton system from actual polygraph examinations. Tables 1 and 2 list the contents of these. In a separate file, a list of the scores and confirmation status for the subjects was provided. The information from this file was extracted and is shown in Table 3. Note that not all subjects listed in Table 3 are actually present in the .ZIP files. In fact, Table 3 contains 484 subjects, 113 more than the combined total of Tables 1 and 2.

**Table 1. Subject contents of CDE.ZIP**

| | | | | | |
|---|---|---|---|---|---|
| $$8#432F | $$8STE83 | $$8YM66# | $$953CKF | $$9NEZJO | $$9WD#F9 |
| $$8#B1I0 | $$8T4XJ9 | $$8YT7SO | $$95C$XC | $$9NTJSO | $$9WR60O |
| $$8#BL80 | $$8UP553 | $$8Z3WYI | $$95JMHI | $$9NTX5T | $$9Y#PPM |
| $$8#BR40 | $$8UQ#DL | $$8ZQD83 | $$95KT83 | $$9O$844 | $$9YM8QF |
| $$8#D92I | $$8UQU1F | $$8ZZBRO | $$95W8#O | $$9O58BX | $$9YMQ1C |
| $$8#DB1F | $$8V#QD0 | $$9#H6P9 | $$96$4MI | $$9OY2DU | $$9ZCTLI |
| $$8#DCN# | $$8V$6AI | $$9#HSDF | $$97CRVI | $$9QEIXC | $$9ZDLRX |
| $$8#N#GF | $$8V1XL6 | $$9$768L | $$97EZ0L | $$9QEMV6 | $$A0555F |
| $$8#NVQU | $$8V4LLS | $$9$LZEC | $$97RAH# | $$9QGS1C | $$A0H5UC |
| $$8#QT54 | $$8V7P$# | $$9$P9L9 | $$97RCW9 | $$9QVG%# | $$A0Z9JF |
| $$8#QX16 | $$8VG$YU | $$9%122C | $$97TK1C | $$9QVSWI | $$A1LWZX |
| $$8#VA3W | $$8VGG%F | $$90#V4I | $$99%NY9 | $$9R44DQ | $$A2R2O0 |
| $$8$1S9X | $$8VOM#F | $$90427I | $$9AC8UX | $$9R6$2X | $$A34NL9 |
| $$8$1XEX | $$8VV7KL | $$908AQX | $$9E1HBU | $$9R72VI | $$A34SAX |
| $$8$2UQ9 | $$8VVNRC | $$90GF2C | $$9EFUAU | $$9RJ683 | $$A353IR |
| $$8%E2O6 | $$8X$$9O | $$90IJZ3 | $$9EGDEC | $$9RLV6I | $$A3G7O0 |
| $$8%HWE9 | $$8X9QQO | $$90UYTV | $$9F%9K3 | $$9RLVLX | $$A3GZOC |
| $$8%SF%L | $$8X9V0X | $$91#5NB | $$9FYL#3 | $$9SNW$9 | $$A3JA#C |
| $$8%ZID0 | $$8XO#$0 | $$91ZJSK | $$9FYPX# | $$9SO7M% | $$A3JH9I |
| $$8LBP#9 | $$8XONK# | $$939#R6 | $$9GPSA0 | $$9SQ#AL | $$A48J50 |
| $$8MFRW6 | $$8XORIU | $$94MT86 | $$9ILG2G | $$9T1WQF | $$A613MO |
| $$8MWQE6 | $$8XOWO4 | $$94RO36 | $$9IP4FR | $$9TG9VI | $$A6GKLL |
| $$8PVLYR | $$8XW5VX | $$94RRJO | $$9IZJKR | $$9TGHR6 | $$A7AA36 |
| $$8QMT#S | $$8Y1FCC | $$94TKJX | $$9J1S36 | $$9TSY4I | $$A9B22C |
| $$8SFFWC | $$8Y7CY3 | $$94TW01 | $$9LKL1Q | $$9U41E# | $$AA#EL3 |
| $$8SFK6L | $$8YDTEC | $$94TZYX | $$9LZW5O | $$9V$FXX | $$ABQ%$O |
| $$8SHYX# | $$8YG14X | $$9537UU | $$9MBH3O | $$9VB3H3 | $$ADYJ3F |

Table 2. Subject contents of ZCT.ZIP

| | | | | | |
|---|---|---|---|---|---|
| $$4#51FR | $$5LFP1X | $$6X#%JU | $$7J$JY0 | $$84XPS0 | $$8H7ARF |
| $$4#LJNX | $$5S5QD9 | $$6X#Y6L | $$7J8L#U | $$85D6NX | $$8HM8DI |
| $$4$%PT9 | $$5SYJNJ | $$6X%P%O | $$7JQOG6 | $$85O2I3 | $$8HO#6G |
| $$4%EL10 | $$5SYNL9 | $$6YBQOL | $$7KDYIO | $$86SLR# | $$8HZWC6 |
| $$4%G8JX | $$5SYQY1 | $$6YCHNU | $$7M$Q$% | $$874K69 | $$8JXFXA |
| $$4OLOQ3 | $$5V54$C | $$6Z4%7O | $$7MOWJ# | $$876WIX | $$8K5%K% |
| $$4OLYKR | $$5VJS7C | $$7#0XI# | $$7MQ$T3 | $$87I$XF | $$8K78PC |
| $$4PBMAI | $$5VZOB# | $$7$5M0X | $$7OW8BI | $$87IM30 | $$8K99#O |
| $$4PT3JO | $$5YLH6L | $$7$7KLI | $$7OY9MU | $$87K$6U | $$8KKW8O |
| $$4PTE60 | $$6#NN63 | $$7$JZF9 | $$7PLPM1 | $$87NG2H | $$8KLQTC |
| $$4R%5EO | $$6$EFAI | $$7$YSR3 | $$7PLT4I | $$87ZH4S | $$8LA%YX |
| $$4R%KTO | $$6%19LL | $$7%CJSU | $$7QSEQ7 | $$888RMB | $$8LANZ9 |
| $$4RO870 | $$6%4%Z9 | $$7%Q6QL | $$7R2E4C | $$88A$KX | $$8LFQ#C |
| $$4ROJ8U | $$6%4VVC | $$72ULQ0 | $$7RGZTU | $$89F$J# | $$8M5D3L |
| $$4TWED6 | $$6B67F9 | $$75T%JC | $$7RU%3X | $$8A30BC | $$8MFXDL |
| $$4U5KJI | $$6CPSOF | $$768T%I | $$7S6XE6 | $$8AIL83 | $$8MVL$X |
| $$4UMJ1I | $$6D0$PL | $$78F9AF | $$7TQFNC | $$8AKOXR | $$8MXYB# |
| $$4WWX%9 | $$6DF8XX | $$797NL3 | $$7U4R60 | $$8AL2GX | $$8N5U66 |
| $$4XKUQU | $$6DX2SM | $$797V1F | $$7UG8I3 | $$8AW85R | $$8N7WOO |
| $$4XYAKF | $$6E5KB9 | $$79JHII | $$7UGGDU | $$8AYVZ# | $$8N8A%L |
| $$4Z15V# | $$6EMMTL | $$79K3CR | $$7 | $$8AYX$C | $$8NJUZ9 |
| $$5#FZ7# | $$6FB2## | $$79KKB9 | $$7V%86C | $$8C$JQ0 | $$8NM9TR |
| $$5#G1G4 | $$6G385U | $$79NBU0 | $$7VZUYF | $$8C0OV3 | $$8OAAVF |
| $$51QEU6 | $$6GL6DI | $$7A%ZLL | $$7WTA#U | $$8CRXD3 | $$8OAB%L |
| $$52F60C | $$6GT6U6 | $$7B0OLX | $$7X1JVR | $$8CUIJR | $$8OEKYC |
| $$54$W9F | $$6HYFW0 | $$7BEND0 | $$7X44M# | $$8D3GE# | $$8QY8DL |
| $$55DCZ# | $$6J06TA | $$7BHJ#U | $$7Y$9NO | $$8D6MH# | $$8S1#F0 |
| $$5A7Q#C | $$6O%GEI | $$7CMXC# | $$7YZ9YX | $$8E#9M3 | $$8SG7NC |
| $$5A9GWT | $$6ONKB# | $$7DQ5P0 | $$7ZDCU3 | $$8EN%XC | 16C9DC24 |
| $$5FJPSC | $$6OPUQ3 | $$7GBT8L | $$7ZEGU0 | $$8FB#S# | 18071708 |
| $$5FVD0M | $$6QJZQU | $$7GC5%# | $$7ZPFUC | $$8FDY3F | 18073560 |
| $$5G8X5K | $$6SQP8R | $$7GDJI3 | $$81#YOX | $$8FE%IR | 18074A78 |
| $$5L18GX | $$6T#RWI | $$7HD$CI | $$831SP3 | $$8FRECI | 1849A5F8 |
| $$5L3I3V | $$6TL#Y# | $$7HG4DF | $$84JAER | $$8H6S09 | 1861A4F0 |

**Table 3. Subject file listing. The trailing 0s and 1s indicate the score (0=not guilty, 1=guilty) and the confirmation status (0=not confirmed, 1=confirmed), respectively.**

| | | | | |
|---|---|---|---|---|
| $$4#51FR 0 0 | $$7%9UOC 1 0 | $$8A30BC 0 0 | $$91ZJSK 0 0 | $$ABPAMB 0 0 |
| $$4#LJNX 0 1 | $$7%CJSU 0 0 | $$8AIL83 0 0 | $$939#R6 0 0 | $$ABQ%$O 1 1 |
| $$4$%PT9 0 0 | $$7%Q6QL 0 0 | $$8AKOXR 0 0 | $$94MT86 1 0 | $$ABR3UE 3 0 |
| $$4%EL10 0 0 | $$70NULL 0 0 | $$8AL2GX 1 0 | $$94RRJO 0 0 | $$ABSPPR 0 0 |
| $$4%G8JX 0 0 | $$72ULQ0 0 0 | $$8ALRWL 0 1 | $$94TKJX 1 0 | $$ACHJAD 3 0 |
| $$4OLYKQ 0 0 | $$73J$TF 0 0 | $$8AW85R 1 0 | $$94TZYX 0 0 | $$ACHMG9 1 1 |
| $$4OLYKR 0 0 | $$75T%JC 1 1 | $$8AYVZ# 0 1 | $$953CKF 0 0 | $$AD$9UF 3 0 |
| $$4PBMAI 0 0 | $$768T%I 1 0 | $$8AYX$C 0 0 | $$95C$XC 0 0 | $$ADME49 3 0 |
| $$4PTE60 0 0 | $$76PD5I 0 0 | $$8C$JQ0 0 1 | $$95JMHI 0 0 | $$ADYJ3F 1 1 |
| $$4R%5EO 1 1 | $$78F9AF 0 0 | $$8C0OV3 1 1 | $$95KT83 1 0 | $$ADYWYX 0 0 |
| $$4R%KTO 1 1 | $$78F9AF 1 0 | $$8CRXD3 1 0 | $$95W8#O 1 0 | $$AEB7CC 1 0 |
| $$4ROJ8U 0 1 | $$797V1F 0 0 | $$8CT%M9 0 0 | $$96$4MI 1 0 | $$AEOEWH 3 0 |
| $$4TUZKU 0 0 | $$79JHII 0 0 | $$8CUIJR 1 0 | $$97CRVI 1 0 | $$AEOIAC 1 1 |
| $$4TWED6 0 0 | $$79K3CR 1 1 | $$8D3GE# 1 1 | $$97EZ0L 1 0 | $$AEOXPC 3 0 |
| $$4U5KJI 1 1 | $$79KKB9 1 1 | $$8D6MH# 0 0 | $$97RAH# 1 0 | $$AEQJ69 1 0 |
| $$4UMJ1I 1 1 | $$79NBU0 1 0 | $$8DHRUF 0 0 | $$97RCW9 0 0 | $$AEQZD0 3 0 |
| $$4WUT5I 0 0 | $$7A%ZLL 1 0 | $$8E#9M3 0 0 | $$97TK1C 1 1 | $$AER0AX 3 0 |
| $$4WWX%9 0 0 | $$7B0OLX 0 0 | $$8EN%XC 0 0 | $$99%NY9 1 1 | $$AERK03 1 1 |
| $$4XKUQU 1 1 | $$7BEND0 1 1 | $$8FB#S# 0 0 | $$9AC8UX 0 0 | $$AF4ST0 3 1 |
| $$4XYAKF 0 1 | $$7BHJ#U 0 0 | $$8FDY3F 0 0 | $$9E1HBU 1 1 | $$AFF4IO 1 1 |
| $$4Z0FA3 1 0 | $$7CMXC# 1 0 | $$8FE%IR 1 1 | $$9EFUAU 1 1 | $$AFH8KO 1 1 |
| $$4Z15V# 1 0 | $$7DQ5P0 0 0 | $$8FRECI 0 0 | $$9EGDEC 1 1 | $$AG#O9F 0 0 |
| $$4ZFOCC 0 1 | $$7GBT8L 1 0 | $$8FTMCC 0 0 | $$9F%9K3 0 0 | $$AG4$KL 3 0 |
| $$5#G1G4 1 0 | $$7GC5%# 0 0 | $$8H6S09 1 1 | $$9FYPX# 1 1 | $$AG5AE3 0 0 |
| $$51QEU6 0 0 | $$7GDJI3 1 0 | $$8H7ARF 0 0 | $$9GPSA0 0 0 | $$AH9%QU 1 0 |
| $$52F60C 1 1 | $$7HD$CI 1 0 | $$8HM8DI 0 0 | $$9ILG2G 0 0 | $$AH9Z#3 1 0 |
| $$54$W9F 0 0 | $$7HG4DF 1 1 | $$8HO#6G 0 1 | $$9IP4FR 1 1 | $$AHO8CL 0 0 |
| $$55DCZ# 0 0 | $$7IXC2C 3 0 | $$8HZWC6 0 0 | $$9IZJKR 1 1 | $$AITAP# 0 0 |
| $$58D0QC 0 0 | $$7IZ9FU 0 1 | $$8JXFXA 0 0 | $$9J1S36 0 0 | $$AJ5K$C 1 0 |
| $$5A7Q#C 0 0 | $$7J$JY0 0 0 | $$8K5%K% 0 0 | $$9LKL1Q 1 1 | $$AJJXJ# 0 1 |
| $$5A9GWT 0 0 | $$7J%A2I 1 1 | $$8K78PC 0 0 | $$9LZW5O 1 0 | $$AJX3YF 1 1 |
| $$5CIVSU 0 0 | $$7J8L#U 0 0 | $$8K99#O 0 0 | $$9MBH3O 1 1 | $$AJY1$N 0 0 |
| $$5FJPSC 0 0 | $$7JPR4U 1 0 | $$8KKW8O 1 1 | $$9NEZJO 1 0 | $$ALSL73 0 0 |
| $$5FVD0M 0 0 | $$7JQOG6 1 1 | $$8KLQTC 0 0 | $$9NTJSO 0 1 | $$ALVNRO 1 1 |
| $$5G8X5K 0 0 | $$7KDYIO 1 0 | $$8LA%YX 1 0 | $$9NTX5T 0 0 | $$AM49TT 1 0 |
| $$5L3I3V 0 0 | $$7LJ7B9 0 0 | $$8LANZ9 1 0 | $$9O$844 1 0 | $$AM6VLO 1 0 |

| | | | | |
|---|---|---|---|---|
| $$5LFP1X 0 0 | $$7M$Q$% 1 1 | $$8LBP#9 0 0 | $$9O58BX 1 1 | $$AMJALI 1 0 |
| $$5LFPIX 0 0 | $$7MAU7C 0 0 | $$8LD%PR 1 1 | $$9OY2DU 0 0 | $$AMX4%F 1 0 |
| $$5S$O$H 1 1 | $$7MOWJ# 1 0 | $$8LFQ#C 1 0 | $$9QEIXC 1 0 | $$AMX6$C 0 0 |
| $$5S5QD9 0 0 | $$7MQ$T3 0 1 | $$8M5D3L 0 0 | $$9QEMV6 1 0 | $$AN%8J# 0 0 |
| $$5SL%99 0 0 | $$7OW8BI 0 0 | $$8MFRW6 0 1 | $$9QGS1C 0 0 | $$AOUOTL 1 1 |
| $$5SYQY1 1 1 | $$7OY9MU 1 0 | $$8MFXDL 1 0 | $$9QHIO6 3 0 | $$AQMQZF 0 0 |
| $$5V54$C 1 0 | $$7PLT4I 0 1 | $$8MGYMU 1 0 | $$9QVG%# 0 0 | $$AQNYX6 0 0 |
| $$5VJS7C 1 0 | $$7QSEQ7 0 0 | $$8MVL$X 1 0 | $$9QVSWI 1 0 | $$AQP4MR 0 0 |
| $$5VZOB# 0 0 | $$7R2E4C 0 0 | $$8MWQE6 1 1 | $$9R44DQ 0 0 | $$ARFJ#F 1 1 |
| $$5Y4UI3 1 0 | $$7R57$F 1 0 | $$8MXYB# 1 1 | $$9R6$2X 1 0 | $$ARU6KR 0 0 |
| $$5YILDL 1 0 | $$7RGZTU 0 1 | $$8N5U66 0 0 | $$9R72VI 1 1 | $$AS3$#O 0 0 |
| $$5YLH6L 1 1 | $$7RJ4TO 0 1 | $$8N7WOO 0 0 | $$9RJ683 0 0 | $$AS3$#O 3 0 |
| $$6#NN63 1 0 | $$7RU%3X 1 1 | $$8N8A%L 0 0 | $$9RLV6I 0 0 | $$ASKOP3 0 0 |
| $$6#ZZNX 0 0 | $$7S6XE6 1 1 | $$8NJUZ9 0 0 | $$9RLVLX 0 0 | $$AUD5L9 1 1 |
| $$6#ZZNX 1 0 | $$7TQFNC 1 0 | $$8NM9TR 0 1 | $$9SNW$9 0 0 | $$AURNUS 3 0 |
| $$6$16BF 0 0 | $$7U4R60 0 0 | $$8OAAVF 1 0 | $$9SO7M% 1 0 | $$AUSM4U 1 1 |
| $$6$EFAI 0 0 | $$7UG8I3 1 0 | $$8OAB%L 0 0 | $$9SQ#AL 0 0 | $$AUT#ER 1 0 |
| $$6$FJCP 0 0 | $$7UGGDU 0 0 | $$8OEKYC 1 1 | $$9T1WQF 0 0 | $$AW7VIC 0 0 |
| $$6$H6DX 0 1 | $$7UV1T# 0 0 | $$8PVLYR 1 1 | $$9TG9VI 0 0 | $$AZ59MX 0 0 |
| $$6%19LL 1 1 | $$7V%86C 0 0 | $$8QY8DL 0 0 | $$9TGHR6 0 0 | $$B1VZ6C 1 0 |
| $$6%4VVC 0 1 | $$7V%XVC 0 0 | $$8S1#F0 1 1 | $$9TSY4I 1 0 | $$B26I#X 0 0 |
| $$66CDJI 0 0 | $$7VZUYF 0 0 | $$8SFK6L 0 0 | $$9U41E# 1 0 | $$B2NWXX 1 0 |
| $$6B67F9 0 0 | $$7WE0B0 0 0 | $$8SG7NC 0 0 | $$9V$FXX 0 0 | $$B3%RXC 1 0 |
| $$6CPSOF 0 0 | $$7WTA#U 1 0 | $$8SHYX# 0 1 | $$9VB3H3 1 1 | $$B55TIO 1 1 |
| $$6CRC74 0 0 | $$7X1JVR 0 0 | $$8STE83 0 0 | $$9WD#F9 1 1 | $$B6CLA6 1 1 |
| $$6D0$PL 0 0 | $$7X44M# 1 1 | $$8UP553 1 1 | $$9WR60O 0 0 | $$B6O5SC 1 1 |
| $$6DF8XX 0 0 | $$7Y$9NO 0 0 | $$8UQU1F 1 0 | $$9Y#PPM 0 0 | $$B6P30R 1 0 |
| $$6DX2SM 1 1 | $$7YZ9YX 0 0 | $$8V$6AI 0 0 | $$9YM8QF 1 0 | $$B72T4L 1 1 |
| $$6E5KB9 0 0 | $$7ZDCU3 1 0 | $$8V1XL6 1 0 | $$9YMQ1C 0 0 | $$B7TLIU 1 1 |
| $$6EMMTL 1 0 | $$7ZEGU0 1 1 | $$8V7P$# 0 0 | $$9Z00JX 0 1 | $$B9%MM0 1 0 |
| $$6FB2## 1 1 | $$7ZPFUC 0 0 | $$8VGG%F 0 0 | $$9ZCTLI 0 0 | $$B9MT83 0 0 |
| $$6G385U 1 1 | $$8#432F 1 0 | $$8VOM#F 1 0 | $$9ZDLRX 0 0 | $$B9O9N0 0 0 |
| $$6GL6DI 0 1 | $$8#B1I0 0 0 | $$8VV7KL 1 0 | $$A0555F 0 0 | $$B9Q8ZF 1 0 |
| $$6GT6U6 0 0 | $$8#BR40 1 1 | $$8VVNRC 1 0 | $$A0H5UC 1 0 | $$B9R86R 0 0 |
| $$6HYFW0 1 1 | $$8#DB1F 1 0 | $$8X9V0X 0 0 | $$A0V%IL 3 0 | $$BA08VO 0 0 |
| $$6J06TA 0 0 | $$8#DCN# 1 1 | $$8XOWO4 0 0 | $$A0W7EF 1 1 | $$BBXOY3 1 1 |
| $$6LDYHL 1 1 | $$8#N#GF 1 1 | $$8XW5VX 1 0 | $$A0X$FL 3 0 | $$BBXQ29 0 0 |
| $$6NIVU0 0 0 | $$8#QX16 0 0 | $$8Y1FCC 1 0 | $$A0Y1$0 0 0 | $$BC%9I6 1 1 |
| $$6NXFNO 1 1 | $$8#VA3W 1 0 | $$8Y7CY3 1 0 | $$A0Z9JF 0 0 | $$BC9FG9 0 0 |

| | | | | |
|---|---|---|---|---|
| $$6O%GEI 0 0 | $$8$1XEX 1 0 | $$8YDTEC 1 0 | $$A1LWZX 0 0 | $$BCBF50 1 0 |
| $$6ONKB# 0 0 | $$8%E2O6 1 0 | $$8YG14X 0 0 | $$A2R2O0 1 0 | $$BCBKA0 1 1 |
| $$6OPUQ3 0 0 | $$8%HWE9 1 1 | $$8YM66# 1 0 | $$A34NL9 1 1 | $$BCNG6C 1 0 |
| $$6QJZQU 1 0 | $$8%SF%L 0 0 | $$8YT7SO 1 1 | $$A34SAX 0 0 | $$BD0V8F 1 1 |
| $$6SQP8R 0 0 | $$8%ZID0 0 0 | $$8ZQD83 1 1 | $$A353IR 0 0 | $$BHKV4R 1 0 |
| $$6T#RWI 1 1 | $$81#YOX 1 1 | $$8ZZBRO 0 1 | $$A3G7O0 0 0 | $$BI7%WQ 0 0 |
| $$6TL#Y# 1 0 | $$831SP3 1 1 | $$9#H6P9 0 0 | $$A3GZOC 1 0 | $$BIB730 1 0 |
| $$6W58T0 0 0 | $$84JAER 1 0 | $$9#HNTM 3 0 | $$A3JA#C 0 0 | $$BK6OER 0 0 |
| $$6X#%JU 1 1 | $$84XPS0 0 0 | $$9#HSDF 0 0 | $$A3JH9I 1 0 | $$BO6URU 0 0 |
| $$6X#Y6L 0 0 | $$85D6NX 1 0 | $$9#WJA# 3 0 | $$A48J50 1 0 | $$BONX7O 0 0 |
| $$6X%P%O 0 0 | $$85O2I3 0 0 | $$9#WN8U 1 1 | $$A613MO 0 0 | $$BQTW#I 1 1 |
| $$6Y02U0 0 0 | $$86SLR# 0 0 | $$9$768L 1 0 | $$A6GKLL 0 0 | $$BQUW%L 0 0 |
| $$6YBQOL 0 0 | $$874K69 0 0 | $$9$LZEC 0 0 | $$A6TX33 0 1 | $$BQVPLC 1 1 |
| $$6YCHNU 0 1 | $$876WIX 0 0 | $$9$P9L9 1 0 | $$A6V8SF 1 1 | $$BQVZF# 0 0 |
| $$6YG5N6 0 0 | $$87I$XF 0 0 | $$9%122C 0 0 | $$A7AA36 0 0 | $$BR5EV6 1 1 |
| $$6Z4%7O 1 0 | $$87IM30 0 0 | $$90#V4I 1 1 | $$A8S3PC 1 1 | $$BSQKTI 0 0 |
| $$6ZG%H6 0 0 | $$87K$6U 0 0 | $$90427I 1 0 | $$A91E#O 3 0 | $$BTI%%0 0 0 |
| $$7#0XI# 0 0 | $$87NG2H 1 1 | $$908AQX 0 0 | $$A9B22C 1 1 | $$BTV8Z4 1 0 |
| $$7$5M0X 0 0 | $$87ZH4S 1 1 | $$90GF2C 1 0 | $$A9F#7R 0 0 | 16C9DC24 0 0 |
| $$7$7KLI 0 0 | $$888RMB 0 1 | $$90IJZ3 0 0 | $$AA#EL3 1 0 | 18074A78 0 0 |
| $$7$JZF9 1 1 | $$88A$KX 0 0 | $$90UYTV 1 0 | $$AA$UEL 0 0 | 1849A5F8 1 1 |
| $$7$YSR3 0 0 | $$89F$J# 1 1 | $$91#5NB 1 1 | $$ABB8%0 0 0 | |

These polygraph examinations generally follow the control / relevant test format with the standard order for questions / events shown in Table 4. This ordering for events will be referred to as the "standard order". However, there were many exceptions to this event ordering encountered in the database.

**Table 4. The "Standard Order"**

| Event Name | Event / Question Type |
|---|---|
| TB | Test Begin |
| N | Neutral |
| SR | Sacrifice Relevant |
| S1 | Symptomatic 1 |
| C1 | Control 1 |
| R1 | Relevant 1 |
| C2 | Control 2 |
| R2 | Relevant 2 |
| S2 | Symptomatic 2 |
| C3 | Control 3 |
| R3 | Relevant 3 |
| ET | End Test |

Each subject contained in the compressed .ZIP files represents one or more charts (usually three charts). Each chart has associated with it three files: an event marker file, the raw signal data, and the event / question description file. The event file contains the location of the events / questions described in the event / question description file. These locations are expressed as integer constants, and they give the absolute locations in the raw data file at which an event begins, ends, and where a subject begins a response. The event / question description file can be viewed directly (after it is "unzipped") using a text editor on the PC, but the event and raw data files are stored in binary format according to the Axciton software, and cannot be viewed directly. The raw data file contains four columns of data: column 1 is the GSR signal, column 2 is the Cardio signal, and columns 3 and 4 are the thoracic and abdominal respiration signals, respectively.

From Table 3, the raw database consists of roughly 484 subjects, each consisting of between 1 and 5 charts. About 129 of these subjects' tests are confirmed (i.e. the score was confirmed either through confession or other means). Of the total subjects, 269 were scored as guilty (55.6%). Of the 129 confirmed subjects, 105 were confirmed guilty (81.4%). This is consistent with information provided prior to this effort by NSA experts, indicating a large bias towards guilty cases being confirmed.

Notice again that the subjects listed in Table 3 do not correlate exactly with the contents of Tables 1 and 2. In particular, the following 8 confirmed not guilty subjects are missing from the .ZIP files: $$4ZFOCC, $$6$H6DX, $$7IZ9FU, $$7RJ4TO, $$8ALRWL, $$9Z00JX, $$A6TX33, $$AJJXJ#. There are other confirmed guilty subjects missing from the .ZIP files, but since there is no shortage of confirmed guilty subjects, we have not tabulated these. Because of the relative shortage of confirmed not guilty cases, the loss of the aforementioned 8 subjects is significant and will diminish the extent to which the classification effectiveness of the ANN can be studied.

### *I.2 Decompressing and Extraction of Data*

The first step in creating a training database for the ANN approach was to extract the confirmed cases. Only confirmed cases are used for training in order to avoid error introduced by incorrect scoring by the examiners.

A program was written to selectively "unzip" the confirmed subjects from the .ZIP files on the Bernoulli disk. This program makes use of the commercial software program for the PC called PKUNZIP, the companion program to PKZIP (also a product of PKWare, Inc.). This program is listed and described in Appendix I-A.

Once a subject was unzipped, all the charts for that subject were then temporarily processed into viewable ASCII files using a C program provided by Mr. Chris Pounds of the University of Washington, a former research assistant involved with the processing and analysis of the Axciton data files. This C program apparently originated with Mr. John Harris of the Johns Hopkins Applied Physics Laboratory, and was modified by us, with the direction and assistance of Mr. Pounds, to run in the PC environment. This program reads the three files for a chart, and creates a large ASCII file consisting of the four polygraph signals, and a fifth column indicating the beginning and termination of various events (e.g. 0=begin question / event, 1=end event / question, 2=begin answer to question). A source listing of this program is included in Appendix A. From these ASCII files, new files are created containing the 5 columns of data (the 4 polygraph signals and the event marker column) and stored in binary format in order to save storage space, and the ASCII files are discarded. (The question files are retained.) These two steps, extraction of the ASCII files and creation of the binary files, are accomplished from one program, listed in Appendix I-A.

### *I.3 Viewing the Charts and Generating a Training File*

Once the confirmed charts are stored in binary files of known format, it is necessary to view them one by one and extract the responses to the control and relevant questions. Again, a custom software program was developed for this task, and is listed in Appendix I-A.

As mentioned previously, there is a "standard format" for the polygraph examinations, shown in Table 4. However, many charts deviate from this order. When deviations occur, the software program attempts to determine the nature of the deviation, and label the events accordingly. If this is not possible, the user can intervene (based on reading / editing of the question file) and correlate the events with the markers manually.

Several options are available in terms of displaying the information using this program. The entire chart can be displayed with events marked, any single response can be displayed, or the control - relevant response pairs can be displayed for comparison. Samples of these generated by our custom software are shown in Figures 1, 2, and 3. In Figures 2 and 3, the graduations are one second increments while in Figure 1, they are 5 second increments.

In Figure 1 the signals GSR, Cardio, Thoracic Respiration, and Abdominal Respiration are displayed from top to bottom. The vertical lines indicate the event markers. On the computer screen they are colored to indicate the event type (white = begin question, blue = end question, red = begin response). In the upper left corner, the subject name is shown and the extension .CH2 indicates that it is the second chart for this subject. GUILT=0 indicates that the chart was scored as not guilty (GUILT=1 indicates guilty) and CONFIRM=1 indicates that the score is confirmed. The GSR and respiration signals are very legible but due to the amount of data and the high variability of the Cardio channel, it is difficult to see all of the detail in that channel. However, these details become clear in Figures 2 and 3.

Figure 2 shows the single question and response to the second control question C2. Also shown on this printout is the range (minimum to maximum) of the data reported by the Axciton system. Figure 3 displays a side by side comparison of control question C2 with the next question on the chart, relevant question R2. Here and in Figure 2, the details in the Cardio channel are now very clear. In particular, the dichotic notch is clearly visible. Here also is displayed other relevant data including the number of sample points in each response (the Axciton system samples the signal 30 times per second) and the Control - Relevant (C-R) pair sequence number.

The ability to view the charts in this manner is necessary to insure that the proper data is included into the ultimate ANN input database. For example, viewing the chart will show the presence of movement or improper event sequence (deviation from the standard order).

Once the chart has been viewed and the proper event order determined or verified, the program optionally creates a binary data file containing the subject and chart number, the score determined by the examiner (guilty or not guilty in the issue at hand), whether the score has been confirmed or not, and the C-R question pairs available (these would normally be (C1, R1), (C2, R2), (C3, R3), but some examinations contain only (C1, R1), (C2, R2)). It is these C-R pairs that will be used in the ANN analysis. The ANN database thus consists of the binary files of C-R pairs.

The program creates two binary files associated with each chart. One file contains the actual time series of data for each C-R pair as reported by the Axciton automated system. The other file contains the C-R pairs transformed so that they will be comparable both within the same chart, and also between charts and different subjects. This transformation, referred to as a "robust" transformation, was used by previous researchers to achieve this comparability. See Pounds and Martin (1993b) and Martin and Pounds (1993a). Denoting Y as a chronological listing of the data from one signal from a C-R pair, the transformation is given by

$$Y \leftarrow \frac{Y - med(Y)}{med(|Y - med(Y)|)}$$

where med(Y) denotes the median value of the vector Y. In words, the observations are centered by their median, and scaled by the median absolute deviation from their median. This transformation tends to remove the effects of arbitrary location and scaling imposed by the Axciton system and the examiner during initial calibration.

The conversion to binary format of the database files saves computer storage space and speeds the input and output process, as binary files are smaller than ASCII files containing the same information, and are read via software virtually without the need for translation. For example, a typical ASCII file containing one complete chart uses approximately 325,000 bytes of information. Translating this into a binary file reduces the size to about 100,000 bytes. The corresponding binary file containing the three C-R pairs uses approximately 45,000 bytes. This is a reduction of about 7.2 to 1 which greatly enhances our ability to store and experiment with the database.

### *I.4 Viewing the C-R Pairs*

The last custom software program developed for the data processing phase of this effort is a file viewer for the C-R pairs files that simply displays the three C-R pairs in a selected file along with the other information in the file. This viewer is necessary to verify that the correct data was included in the file, and later to examine the ANN input files in case anomalies arise during analysis. Figure 4 is a sample printout of one of the C-R pairs from a C-R pair file view.

Figure 4 displays the third C-R pair from the chart considered in Figures 1 through 3, but reads the data from the binary C-R pair file created by the database program. Thus, this display can be compared with the screens created by the database program to verify that the correct data has been placed in the database or used later to review a data item that may be noteworthy for some reason. This display also contains data needed by the ANN in order to read the C-R pair file, namely the C-R pair sequence number, the number of samples in each response, the total number of C-R pairs, and the number of channels (four polygraph signals plus one channel for the event markers).

### *I.5 The ANN Training Database*

Because of the nature of the Axciton files, it is virtually impossible to process the Axciton data into usable form in an entirely automated process. Manual viewing of the data and question files is essential to insure integrity of the database. This has been encountered by other researchers in this field. See Martin and Pounds (1993a), and Pounds and Martin (1993b), for example.

As a result of using our custom software for processing the polygraph data, an ANN database consisting of 152 charts was established. These 152 charts are listed in Table 5. It is noted that this represents only a fraction of the confirmed charts from subjects listed in Table 3, a potential yield of roughly 387 charts (129 confirmed subjects times 3 charts per subject). Thus, processing of a large number of confirmed subjects failed.

In general, four causes for the failure to extract a given chart were observed as follows: (1) the Axciton files could not be read by the file extraction program supplied by Mr. Chris Pounds of the University of Washington; (2) the Axciton file containing event markers was corrupted or incomplete; (3) the question file could not be correlated with the events reported for the chart; (4) the chart was in a compressed format even after "unzipping" it from the original .ZIP file.

Of the 152 charts listed in Table 1, 29 (19%) were confirmed as "not guilty" and the remaining 123 were confirmed as "guilty." This bias towards confirmed guilty cases is consistent with the over all database percentage. That is, the entire database supplied by Dr. Olsen contained 129 confirmed subjects, of which 24 were confirmed not guilty (about 18.6%). Thus, the inability to process all subjects and charts in the Axciton database supplied by Dr. Olsen did not change the degree of bias inherent in the original database. However, it is noted here that assuming 3 charts per subject, the original 129 confirmed subjects could have potentially produced about 387 charts, of which roughly 18.6%, or about 72 charts, would have been confirmed not guilty. Obviously, a database containing all 72 of these confirmed not guilty charts would have been more desirable for the present study.

**Table 5. Confirmed Subject Charts for the ANN Experimentation**

| | | | |
|---|---|---|---|
| $$4#ljnx.ch1 | $$6t#rwi.ch3 | $$7zegu0.ch1 | $$8mwqe6.ch1 |
| $$4#ljnx.ch2 | $$6x#%ju.ch1 | $$7zegu0.ch2 | $$8mwqe6.ch2 |
| $$4#ljnx.ch3 | $$6x#%ju.ch2 | $$7zegu0.ch3 | $$8mwqe6.ch3 |
| $$4r%kto.ch1 | $$6x#%ju.ch3 | $$8#br40.ch1 | $$8mxyb#.ch1 |
| $$4r%kto.ch2 | $$6ychnu.ch3 | $$8#br40.ch2 | $$8mxyb#.ch2 |
| $$4r%kto.ch3 | $$6ychnu.ch4 | $$8#br40.ch3 | $$8mxyb#.ch3 |
| $$4roj8u.ch1 | $$7$jzf9.ch1 | $$8#dcn#.ch1 | $$8nm9tr.ch1 |
| $$4u5kji.ch1 | $$7$jzf9.ch2 | $$8#dcn#.ch2 | $$8nm9tr.ch2 |
| $$4u5kji.ch2 | $$7$jzf9.ch3 | $$8#dcn#.ch3 | $$8nm9tr.ch3 |
| $$4u5kji.ch3 | $$75t%jc.ch2 | $$8#n#gf.ch1 | $$8oekyc.ch1 |
| $$4umj1i.ch1 | $$75t%jc.ch3 | $$8#n#gf.ch2 | $$8oekyc.ch2 |
| $$4umj1i.ch2 | $$79k3cr.ch1 | $$8#n#gf.ch3 | $$8oekyc.ch3 |
| $$4umj1i.ch3 | $$79k3cr.ch2 | $$87ng2h.ch1 | $$8s1#f0.ch1 |
| $$4xkuqu.ch1 | $$79k3cr.ch3 | $$87ng2h.ch2 | $$8s1#f0.ch2 |
| $$4xkuqu.ch2 | $$79kkb9.ch1 | $$87ng2h.ch3 | $$8s1#f0.ch3 |
| $$4xkuqu.ch3 | $$79kkb9.ch2 | $$888rmb.ch4 | $$8shyx#.ch1 |
| $$4xyakf.ch1 | $$79kkb9.ch3 | $$89f$j#.ch1 | $$8shyx#.ch2 |
| $$4xyakf.ch2 | $$7bend0.ch1 | $$89f$j#.ch2 | $$8shyx#.ch3 |
| $$4xyakf.ch3 | $$7bend0.ch2 | $$89f$j#.ch3 | $$8up553.ch1 |
| $$52f60c.ch1 | $$7bend0.ch3 | $$8c$jq0.ch1 | $$8up553.ch2 |
| $$52f60c.ch2 | $$7hg4df.ch1 | $$8c$jq0.ch3 | $$8up553.ch3 |
| $$52f60c.ch3 | $$7hg4df.ch2 | $$8c$jq0.ch4 | $$8yt7so.ch1 |
| $$6%19ll.ch1 | $$7hg4df.ch3 | $$8c0ov3.ch1 | $$8yt7so.ch2 |
| $$6%19ll.ch2 | $$7jqog6.ch1 | $$8c0ov3.ch3 | $$8yt7so.ch3 |
| $$6%19ll.ch3 | $$7jqog6.ch2 | $$8c0ov3.ch4 | $$8zqd83.ch1 |
| $$6%4vvc.ch1 | $$7jqog6.ch3 | $$8d3ge#.ch2 | $$8zzbro.ch1 |
| $$6dx2sm.ch1 | $$7m$q$%.ch1 | $$8d3ge#.ch3 | $$8zzbro.ch2 |
| $$6dx2sm.ch2 | $$7m$q$%.ch2 | $$8h6s09.ch1 | $$8zzbro.ch3 |
| $$6dx2sm.ch3 | $$7m$q$%.ch3 | $$8h6s09.ch2 | $$90#v4i.ch1 |
| $$6fb2##.ch1 | $$7mq$t3.ch1 | $$8h6s09.ch3 | $$90#v4i.ch3 |
| $$6fb2##.ch2 | $$7mq$t3.ch2 | $$8ho#6g.ch1 | $$90#v4i.ch4 |
| $$6g385u.ch1 | $$7mq$t3.ch3 | $$8ho#6g.ch2 | $$91#5nb.ch1 |
| $$6g385u.ch4 | $$7plt4i.ch1 | $$8ho#6g.ch3 | $$91#5nb.ch2 |
| $$6hyfwo.ch2 | $$7plt4i.ch2 | $$8kkw8o.ch2 | $$97tk1c.ch1 |
| $$6hyfwo.ch3 | $$7plt4i.ch3 | $$8kkw8o.ch3 | $$97tk1c.ch3 |
| $$6hyfwo.ch4 | $$7rgztu.ch1 | $$8mfrw6.ch1 | $$97tk1c.ch4 |
| $$6t#rwi.ch1 | $$7rgztu.ch2 | $$8mfrw6.ch2 | 1849a5f8.ch1 |
| $$6t#rwi.ch2 | $$7rgztu.ch3 | $$8mfrw6.ch3 | 1849a5f8.ch2 |

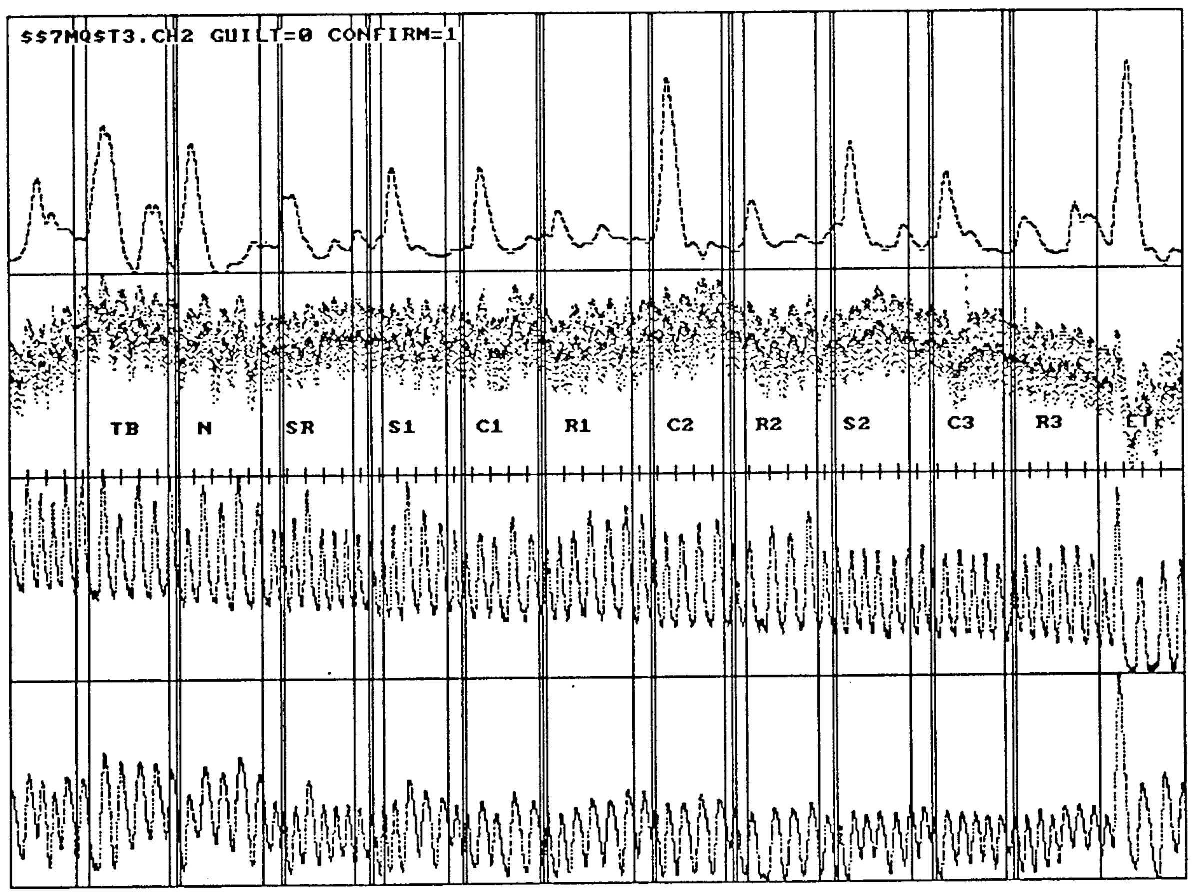



Figure 1. Sample Printout of Whole Chart from Custom Database Software

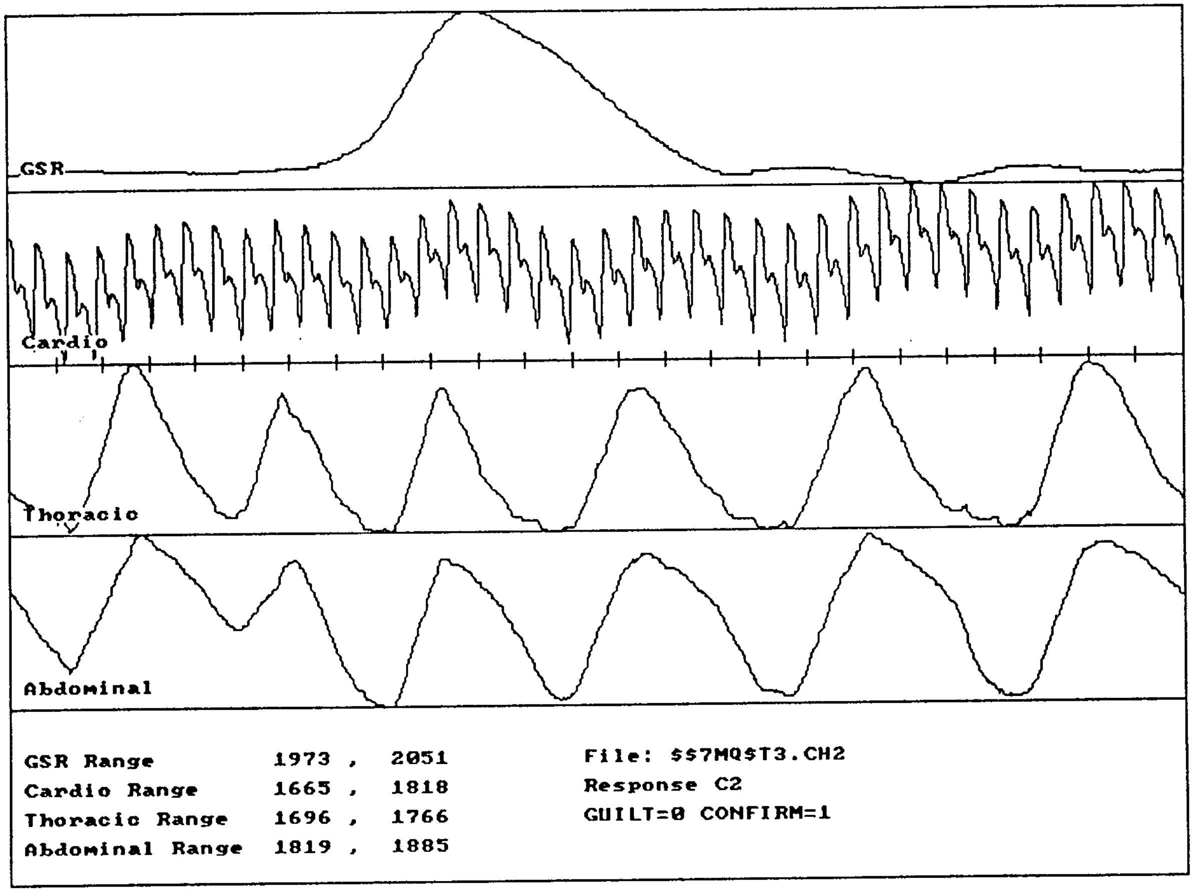



Figure 2. Sample Printout of Control Question C2 Response from Custom Database Software

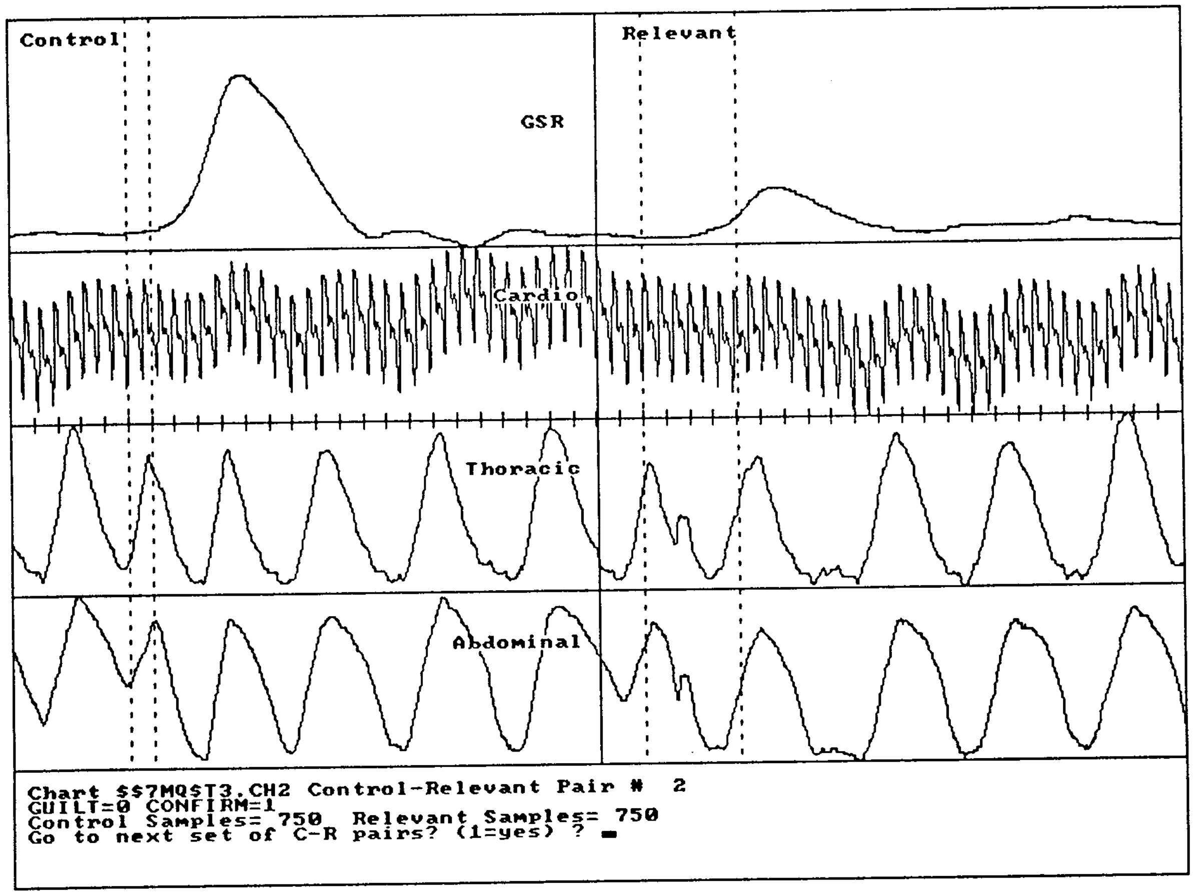



Figure 3. Sample Printout of Control -Relevant Pair (C2, R2) from Custom Database Software

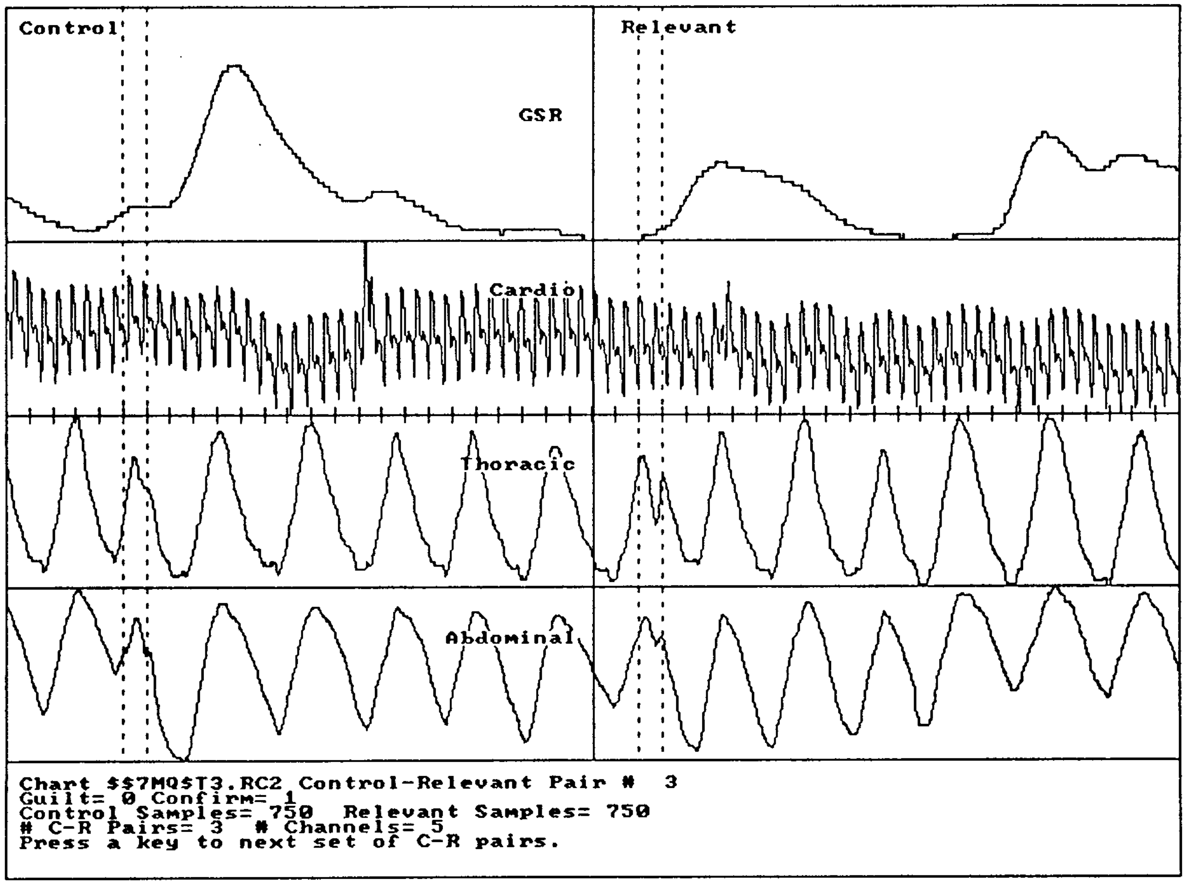



Figure 4. Sample Printout of Control -Relevant Pair (C3, R3) from Custom Database Viewer Software

### *I.6 References for Section I*


Kircher, J.C. and Raskin, D.C. (1988). Human versus computerized evaluations of polygraph data in a laboratory setting. J. Appl. Psych. 73(2), 291-302.

Martin, R.D. and Pounds, C. (1993a). Classification trees and spectral methods applied to polygraph data. University of Washington, research progress report.

Olsen, D.E., Ansley, N, Feldberg, I.E., Harris, J.C., and Cristion, J.A. (1991). Recent developments in polygraph technology. Johns Hopkins APL Technical Digest 12(4), 347-357.

Pounds, C. and Martin, R.D. (1993b). Polygraph reliability: year two report. University of Washington, research progress report.

Raskin, D.C., Kircher, J.C., Honts, J.C., and Horowitz, S.W. (1988). A study of the validity of polygraph examinations in criminal investigations. Final report, National Institute of Justice.

# Appendix I-A

**Software source listings.**

```
'    UNZIPUM.BAS
'
'    This program will select individual subjects from catalogs of the
'    .ZIP files and unzip all the corresponding charts and files.  The
'    file confirm1.txt contains each subject name along with its score
'    and confirmation status.  The files cdecont.lis and zctcont.lis
'    contain the directory listings from CDE.ZIP and ZCT.ZIP, respectively.
'    These are created by running the "filelist" option of PKUNZIP.
'
CLS
INPUT "Enter 0 for all files, 1 for confirmed, and 2 for unconfirmed: "; icase
CLS
OPEN "confirm1.txt" FOR INPUT AS #1
PRINT "Enter a 1 to select a subject to process."
isel = 0
DO WHILE isel <> 1
readanother:
IF EOF(1) THEN
    CLOSE 1
    OPEN "confirm1.txt" FOR INPUT AS #1
END IF
LINE INPUT #1, filerec$
IF icase <> 0 THEN confirm$ = MID$(filerec$, 26, 1)
IF icase = 1 AND confirm$ <> "1" THEN GOTO readanother
IF icase = 2 AND confirm$ <> "2" THEN GOTO readanother
LOCATE 2: PRINT filerec$
INPUT isel
LOOP
LOCATE 3: PRINT filerec$ + " selected."
CLOSE 1
subject$ = MID$(filerec$, 1, 8)
REM  Must find possible directory prefix.
OPEN "zctcont.lis" FOR INPUT AS 2
OPEN "cdecont.lis" FOR INPUT AS 3
found = 0
DO UNTIL EOF(3)
LINE INPUT #3, filename$
subj$ = MID$(filename$, 1, 8)
FOR i = 1 TO LEN(filename$)
IF MID$(filename$, i, 1) = "/" THEN
    subj$ = MID$(filename$, i + 1, 8)
    GOTO foundslash
END IF
NEXT i
IF subj$ = subject$ THEN
```

```
      filepath$ = MID$(filename$, 1, i)
      foundin$ = "cde.zip"
      found = 1
      GOTO foundit
END IF
LOOP
DO UNTIL EOF(2)
LINE INPUT #2, filename$
subj$ = MID$(filename$, 1, 8)
FOR i = 1 TO LEN(filename$)
IF MID$(filename$, i, 1) = "/" THEN
      subj$ = MID$(filename$, i + 1, 8)
      GOTO foundslash
END IF
NEXT i
foundslash:
IF subj$ = subject$ THEN
      filepath$ = MID$(filename$, 1, i)
      foundin$ = "zct.zip"
      found = 1
      GOTO foundit
END IF
LOOP
foundit:
IF found <> 1 THEN
      PRINT "File not found in ZIP files."
      END
END IF
IF i > LEN(filename$) THEN filepath$ = ""
PRINT "Subject is "; subj$; " prefix is "; filepath$
INPUT x
PRINT "Extracting "; filepath$ + subject$ + ".*"
SHELL "pkunzip " + foundin$ + " " + filepath$ + subject$ + ".*"
CLS
SHELL "dir " + subject$ + ".* > files.dir"
CLOSE
END
```

```
/*    NF4.C

      This program was supplied by Mr. Chris Pounds of the University of
      Washington.  It has been significantly modified to run in a DOS
      environment using Borland Turbo C++.                        */

/* This version modified on April 9, 1993 to handle more general file */
/* formats.  The modifications were supplied by Chris Pounds.          */

#include <stdio.h>
#include <math.h>
#include <string.h>


/* unsigned short int m[15000][4]; */
unsigned short int m1[7500][4];
unsigned short int m2[7500][4];

char        *infilename;
char        *outfilename;

FILE        *datfile, *odatfile, *fptr;
int         ninfo[4];
int         nrow, ncol;

char        questions[20][80];
int         ievent[256];

int         eventtype[256];

char        *filename;
unsigned short int idummy = 0;
int         ldummy;
unsigned short int i5[5];
unsigned short int jdummy[3];
int         i, j, k;
int         nact = 0, nskip = 0;
int         nchannels;
char        inversionflag[4];
int         samplerate = 30, lengthevents;
short int   nsec;
short int   nquest;
char        idinfo[236];
char        tempstr[256];
char        errfile[20];
```

```
int          number_items_really_read;


unsigned short int ii, iidummy;

main(argc, argv)
 int          argc;
 char         *argv[];
{
 infilename = argv[1];
 outfilename = argv[2];
 if ((odatfile = fopen(outfilename, "w")) == NULL) {
  printf("problems opening file:\n");
  exit(2);
 }
 for (i = 0; i < 256; i = i + 1) {
  ievent[i] = -1;
  eventtype[i] = 9;
 }

 for (k = 0; k < 64; k++)

  if (infilename[k] == '\0')
   break;
 infilename[k - 1] = '1';
 if (!(fptr = fopen(infilename, "rb"))) {
  perror("fopen");
  exit(23);
 }
 number_items_really_read = fread(idinfo, 1, 236, fptr);
 /* printf(" first read #items = %d \n", number_items_really_read); */

 idinfo[63] = 0;
 idinfo[138] = 0;
 idinfo[146] = 0;
 idinfo[154] = 0;
 idinfo[170] = 0;
 nsec = atoi(idinfo + 134);
 /* printf(" nsec = %d \n", nsec); */

 nchannels = atoi(idinfo + 144);

 /* printf("nchannels= %d \n", nchannels); */

 if (nchannels != 4) {
```

```
 fprintf(errfile, "nchannels=%ld\n", nchannels);
 /* jchexit("Can't handle file format", infilename); */
}
samplerate = atoi(idinfo + 152);
/* printf(" samplerate = %d \n", samplerate); */

nquest = atoi(idinfo + 168);
/* printf("nqust = %d \n", nquest); */

fread(inversionflag + 3, 1, 1, fptr);
for (i = 0; i < 50; i = i + 1)
 fread(inversionflag + 2, 1, 1, fptr);
for (i = 0; i < 50; i = i + 1)
 fread(inversionflag + 1, 1, 1, fptr);
for (i = 0; i < 50; i = i + 1)
 fread(inversionflag + 0, 1, 1, fptr);
/*
 * This was +0,1,2,3 I added 1 to see what would happen byte swapping might
 * make changes to this section necessary
 */

fread(&ldummy, 1, 1, fptr);
/* first file contains marks for question starts */

/* read when each question starts */

for (i = 0; i < 256;) {
  if ( fread( i5, 2, 5, fptr ) == 0 )
              goto eof011;
  /* this part commented out for dos */
  /* read 5 two byte things into i5 */
  /*    for (ii = 0; ii < 5; ii++) {
   if (fread(&iidummy, 2, 1, fptr) == 0) { */
      /*

       * printf(" bad read for question markers file 1 \n");
      goto eof011;
   }
   swab(&iidummy, &iidummy, 2);
   i5[ii] = iidummy;
  }   end of do loop on ii */

  /*
   * i5[1] is a marker how many seconds into time series the question event
   * is
```

```
 */

 switch (i5[0]) {
 case 1:    /* beginning */
  ievent[i] = i5[1] * samplerate;
  eventtype[i] = 0;
  i = i + 1;
  break;
 case 2:    /* end */
  ievent[i] = i5[1] * samplerate;
  eventtype[i] = 1;
  i = i + 1;
  break;
 case 3:    /* void */
  ievent[i] = i5[1] * samplerate;
  eventtype[i] = 3;
  i = i + 1;
  break;
 case 4:    /* yes */
  ievent[i] = i5[1] * samplerate;
  eventtype[i] = 2;
  i = i + 1;
  break;
 case 5:    /* no */
  ievent[i] = i5[1] * samplerate;
  eventtype[i] = 2;
  i = i + 1;
  break;
 }
}
eof011:
 ;
 fclose(fptr);
 /* printf("i = %d \n",i); */
 lengthevents = i;

 /*
  * for (j =0;j<i;j++) { printf("%d \t %d \t \n",ievent[j],eventtype[j]);}
  */
 /* printf("\n"); */

 /* printf(" checking for good infilename \n"); */

 /*
```

```
* if the file name contains a c then it must go to a decompress routine
* which is in dos


*/

/* Check for good infilename */
/* printf(" just before check on 2 \n"); */

infilename[k - 1] = '2';
if (!(fptr = fopen(infilename, "rb"))) {
 infilename[k - 1] = 'C';
 if (!(fptr = fopen(infilename, "rb"))) {
  perror(NULL);
  printf(" cant open %s \n", infilename);
  exit();

 }
}
nact = nsec * samplerate;
/* printf("nact = %d \n", nact); */

/*

 * if (nact*(nchannels+nextra)>(int)DATABUFFERSIZE) makearray("channels",
 * 2, NULL, y, (int) nact, (int) (nchannels+nextra)); else
 * makearray("channels", 2, DATABUFFER, y, (int int) nact, (int)
 * (nchannels+nextra));
 */

/* printf(" reading stuff from 2 file\n"); */

/* for (j = (nchannels - 1); j < nchannels; j = j - 1) { */
 if(inversionflag[0]==120) inversionflag[1]=1;
 for (j = (nchannels - 1); j >= 0; j--) {
 if ((inversionflag[j] != 0) && (inversionflag[j] != 1)) {
/*   fprintf(errfile,
          "Inversion flag= %hd\n", inversionflag[j]);  */
  printf("Can't handle file format inversion %s",
         infilename);
  printf("but we will try anyway.\n");
/*   exit();                          */
 }
 for (i = 0; i < nact; i = i + 1) {
  if (fread(&idummy, 2, 1, fptr) == 0) goto eof012;
```

```
/* swab(&idummy, &idummy, 2); */
  if (j == 1)
       fread(jdummy, 2, 3, fptr);
  if (i<=7500)
/*    if (inversionflag[j] == 0)                    */
        if (inversionflag[j]!=2)
          m1[i][j] = idummy;
        else
          m1[i][j] = -idummy;
  else
/*   if (inversionflag[j] == 0)                    */
        if(inversionflag[j]!=2)
          m2[i-7500][j] = idummy;
        else
          m2[i-7500][j] = -idummy;
 }
 for (i = 0; i < nskip; i = i + 1) {
   fread(&idummy, 2, 1, fptr);
   if (j == 1)
        fread(jdummy, 2, 3, fptr);
 }
}
eof012:
  ;

fclose(fptr);
infilename[k - 1] = '3';
printf(infilename);
/* printf(" doing a read on a 3 file \n"); */
/* Check for good infilename */
if (!(fptr = fopen(infilename, "rt"))) {
  perror(NULL);
  printf("Can't open file %s :", infilename);
}
for (i = 0; i < nquest;) {
  for (j = 0; j < 256; j = j + 1) {
    if (fscanf(fptr, "%c", &tempstr[j]) == EOF)
         goto eof013;
    if (tempstr[j] == '\n')
         break;
  }
  if (j >= 256)
    /* jchexit("Bad file:",infilename */
    if ((tempstr[0] != ' ') || (tempstr[1] != ' ') || (tempstr[2] != ' ')
         || (tempstr[3] != ' ')) {
```

```
        strncpy(&questions[i][0], tempstr, 79);
        questions[i][79] = '\0';
        i = i + 1;
    }
  }
eof013:
  ;

  fclose(fptr);
  /* ninfo replaces:  nchan,samrate,nact2, nextra; in this order */
  ninfo[3] = nact;
  ncol = nchannels;
  ninfo[1] = ncol;

  ninfo[2] = samplerate * ncol;

  for (i = k; i < 13; i = i + 1)
    infilename[i] = ' ';
  for (i = 0; i < 8; i = i + 1)

    infilename[i + 13] = idinfo[i];
  infilename[21] = ' ';
  for (i = 9; i < 17; i = i + 1)
    infilename[i + 13] = idinfo[i - 1];
  infilename[30] = ' ';
  for (i = 18; i < 23; i = i + 1)
    infilename[i + 13] = idinfo[i - 2];
  infilename[36] = ' ';
  for (j = 37, i = 21; ((i < 200)
                        && (j < 63)); i = i + 1)
    if (idinfo[i - 1] != ' ' || idinfo[i] != ' ') {
      infilename[j] = idinfo[i];
      j = j + 1;
    }
  infilename[63] = 0;
  /* endo fo old readax */
  /*
   * printf("dumping ninfo[3] = %d \n", ninfo[3]); printf("dumping ninfo[1] =
   * %d \n", ninfo[1]);
   */

  /* for the length of the time series */
  k = 0;
  for (i = 0; i < ninfo[3]; i++) {
```

```
 /* for each channel */
if (i<=7500)
 for (j = 0; j < ninfo[1]; j++) {
  fprintf(odatfile, "%6d ", m1[i][j]);
  }
else
  for (j = 0; j < ninfo[1]; j++) {
  fprintf(odatfile, "%6d ", m2[i-7500][j]);
 }
      /* adding the fifth column that will print the inversion flags */□
 if (i < ievent[lengthevents - 1]) {
  if (i > ievent[k] - 1) {
      fprintf(odatfile, "%1d ", eventtype[k]);
      k++;
   } else
      fprintf(odatfile, "9 ");

 } else
  fprintf(odatfile, "9 ");
 fprintf(odatfile, "\n");
}

/* return(0); */
}/* end of program */
```

```
'    ASCTOBIN.BAS
'
'    This program selects files that have been unzipped from both CDE.ZIP
'    and ZCT.ZIP, runs NF4.C to create an ASCII file of the chart, and then
'    produces a binary file in known format for later use.
'
TYPE binrec
     gsr AS INTEGER
     cardio AS INTEGER
     thoracic AS INTEGER
     abdominal AS INTEGER
     event AS INTEGER
END TYPE
DIM rec AS binrec
CLS
INPUT "Enter access type: 0=All Files, 1=Confirmed only, 2= Unconfirmed only "; iacc
SHELL "dir > axciton.dir"
OPEN "axciton.dir" FOR INPUT AS #1
OPEN "history.txt" FOR APPEND AS #7
PRINT #7, "******** "; DATE$; " ******** "; TIME$
readit: LINE INPUT #1, filerec$
subject$ = MID$(filerec$, 1, 8)
k$ = MID$(filerec$, 12, 1)
OPEN "confirm.txt" FOR INPUT AS #5
DO UNTIL EOF(5)
LINE INPUT #5, confirmrec$
sub$ = MID$(confirmrec$, 1, 8)
IF sub$ = subject$ THEN
     conf$ = MID$(confirmrec$, 26, 1)
     guilt$ = MID$(confirmrec$, 16, 1)
     GOTO foundit
END IF
LOOP
foundit: CLOSE 5
IF iacc = 0 THEN GOTO continue
IF iacc = 1 AND conf$ <> "1" THEN GOTO readit
IF iacc = 2 AND conf$ <> "0" THEN GOTO readit
continue:
IF k$ = "3" THEN
     quesfile$ = MID$(filerec$, 1, 8) + "." + MID$(filerec$, 10, 3)
     SHELL "copy " + quesfile$ + " d:\polygrap\database"
END IF
IF k$ <> "1" AND k$ <> "3" THEN
     IF EOF(1) THEN GOTO terminate
     GOTO readit
```

```
END IF
IF k$ = "1" THEN
     name$ = MID$(filerec$, 1, 8)
     chart$ = MID$(filerec$, 11, 1)
     file$ = MID$(filerec$, 1, 12)
     MID$(file$, 9, 1) = "."
     PRINT "Processing subject "; subject$; " guilt="; guilt$; ", confirm="; conf$
     PRINT "Processing "; file$
     PRINT #7, file$; " "; "GUILT="; guilt$; " "; "CONFIRM="; conf$
     SHELL "nf4 " + file$ + " wazu.out"
     PRINT
     PRINT "AXCITON files converted. Now creating a binary file."
     PRINT "Opening "; name$ + ".ch" + chart$
     OPEN "d:\polygrap\database\" + name$ + ".ch" + chart$ FOR RANDOM AS #3 LEN = 10
     OPEN "wazu.out" FOR INPUT AS #2
     count = 0
     DO UNTIL EOF(2)
     INPUT #2, rec.gsr, rec.cardio, rec.thoracic, rec.abdominal, rec.event
     count = count + 1
     PUT #3, count, rec
     LOOP
     PRINT count; " records written into "; name$ + ".ch" + chart$
     END IF
     CLOSE #2
     CLOSE #3
IF EOF(1) THEN GOTO terminate
GOTO readit
terminate:
CLOSE
KILL "wazu.out"
END
```

```
'   DATA2.BAS
'
'   This program works directly from the binary
'   files created either from the program ASCTOBIN.BAS.  The binary
'   files all have the extension .CH*, where * is the chart number.
'   It allows viewing and editing of the question file, and extraction
'   of the C-R pairs.
'
'   The programs list.com and te.exe are file viewing and editing programs,
'   respectively, that are available as public domain software.
'
DEFINT I-N
DECLARE SUB robust (arrayin!(), arrayout!(), n AS INTEGER, arraymed!, devmed!)
DECLARE SUB sort (arrayin!(), arrayout!(), n AS INTEGER)
DECLARE FUNCTION xmax! (x!, y!)
DECLARE FUNCTION xmin! (x!, y!)
DIM observedevent$(20)
DIM option$(4)
DIM filet AS STRING * 12
DIM filer AS STRING * 12
DIM startc AS INTEGER, startr AS INTEGER
DIM finishc AS INTEGER, finishr AS INTEGER
DIM gs(2500)
DIM card(2500)
DIM thor(2500)
DIM abdomin(2500)
DIM gst(2500)
DIM cardt(2500)
DIM thort(2500)
DIM abdomint(2500)
DIM event(2500) AS INTEGER
DIM defaultloc(20), indexc(6) AS SINGLE, indexr(6) AS SINGLE
TYPE binrec
    gsr AS INTEGER
    cardio AS INTEGER
    thoracic AS INTEGER
    abdominal AS INTEGER
    event AS INTEGER
END TYPE
TYPE floatrec
    gsr AS SINGLE
    cardio AS SINGLE
    thoracic AS SINGLE
    abdominal AS SINGLE
    event AS INTEGER
```

```
END TYPE
DIM rec AS binrec, recf AS floatrec
DIM rec1 AS binrec
DIM rec2 AS binrec
DIM zeros(20) AS INTEGER, ones(20) AS INTEGER, twos(20) AS INTEGER
DIM eventstr(20) AS STRING * 2
1000 :
FOR jj = 1 TO 20
zeros(jj) = 0
ones(jj) = 0
twos(jj) = 0
NEXT jj
CLS
SHELL "dir > files.dir"
OPEN "files.dir" FOR INPUT AS #9
OPEN "charts.dir" FOR OUTPUT AS #10
DO UNTIL EOF(9)
LINE INPUT #9, record$
ext$ = MID$(record$, 10, 2)
ext1$ = MID$(record$, 10, 1)
ext2$ = MID$(record$, 12, 1)
IF ext$ = "CH" THEN□
    filerec$ = MID$(record$, 1, 12)
    MID$(filerec$, 9, 1) = "."
    PRINT #10, filerec$
END IF
LOOP
CLOSE
PRINT "Select a file to process (i.e. a .ch* file)."
PRINT "1 Selects a file, 0 continues to the next file: "
restart: OPEN "charts.dir" FOR INPUT AS #3
continue: INPUT #3, filerec$
subject$ = MID$(filerec$, 1, 8)
OPEN "confirm.txt" FOR INPUT AS #4
ifound = 0
DO UNTIL EOF(4)
    LINE INPUT #4, scoreandconfirm$
    name$ = MID$(scoreandconfirm$, 1, 8)
    IF name$ = subject$ THEN
        ifound = 1
        GOTO found
    END IF
LOOP
found: CLOSE 4
IF ifound = 1 THEN
```

```
        score$ = MID$(scoreandconfirm$, 10, 7)
        confirm$ = MID$(scoreandconfirm$, 18, 9)
ELSE
        score$ = "Score not found"
        confirm$ = "Confirmation not found"
END IF
LOCATE 3, 1: PRINT SPACE$(78)
LOCATE 3, 1: PRINT filerec$; " was "; score$; " , "; confirm$
INPUT iselect
IF iselect = 0 THEN
        IF EOF(3) THEN
                CLOSE #3
                GOTO restart
        END IF
        GOTO continue
ELSE
        GOTO endfileselect
END IF
endfileselect:
file$ = filerec$
PRINT file$; " selected."
OPEN file$ FOR RANDOM AS #1 LEN = 10
count = 0
izero = 0
ione = 0
itwo = 0
c1max = 0: c1min = 2 ^ 16 - 1: c2max = 0: c2min = c1min: c3max = 0: c3min = c1min: c4max =
0: c4min = c1min
readit: count = count + 1
GET #1, count, rec
IF EOF(1) THEN GOTO endread
y1 = rec.gsr: y2 = rec.cardio: y3 = rec.thoracic: y4 = rec.abdominal
c1max = xmax(c1max, y1): c1min = xmin(c1min, y1)
c2max = xmax(c2max, y2): c2min = xmin(c2min, y2)
c3max = xmax(c3max, y3): c3min = xmin(c3min, y3)
c4max = xmax(c4max, y4): c4min = xmin(c4min, y4)
IF rec.event <> 9 THEN
        IF rec.event = 0 THEN
                izero = izero + 1
                zeros(izero) = count
                END IF
        IF rec.event = 1 THEN
                ione = ione + 1
                ones(ione) = count
                END IF
```

```
    IF rec.event = 2 THEN
        itwo = itwo + 1
        twos(itwo) = count
        END IF
    END IF
GOTO readit
endread: count = count - 1
REM PRINT c1min; c1max; c2min; c2max; c3min; c3max; c4min; c4max
CLS
PRINT "There are "; count; "records in the file."
PRINT
PRINT "There are "; izero; "0 event markers."
IF zeros(13) = 0 THEN zeros(13) = count
PRINT "They occur at record positions:"
FOR i = 1 TO izero
PRINT zeros(i);
NEXT i
PRINT
PRINT
PRINT "There are "; ione; "1 event markers."
PRINT "They occur at record positions:"
FOR i = 1 TO ione
PRINT ones(i);
NEXT i
PRINT
PRINT
PRINT "There are "; itwo; "2 event markers."
PRINT "They occur at record positions:"
FOR i = 1 TO itwo
PRINT twos(i);
NEXT i
PRINT
PRINT
ques$ = file$
MID$(ques$, 10, 1) = "0"
MID$(ques$, 12, 1) = "3"
k$ = MID$(file$, 12, 1)
MID$(ques$, 11, 1) = k$
SHELL "copy standard.ord+" + ques$ + " question.tmp"
standardorder$ = "(X)TB (1)N (2)SR (3)S1 (4)C1 (5)R1 (6)C2 (7)R2 (8)S2 (9)C3 (10)R3
(XX)END"
INPUT "Want to view the question file to check event order? (1=yes, 0=no) "; iquest
IF iquest = 1 THEN
    SHELL "list question.tmp"
END IF
```

```
INPUT "Want to edit the question file? (1=yes, 0=no) "; iedit
IF iedit = 1 THEN
      SHELL "te " + ques$
END IF
CLS
eventstr(1) = "TB": eventstr(2) = "N": eventstr(3) = "SR": eventstr(4) = "S1": eventstr(5) = "C1"
eventstr(6) = "R1": eventstr(7) = "C2": eventstr(8) = "R2": eventstr(9) = "S2": eventstr(10) =
"C3":
eventstr(11) = "R3"
eventstr(12) = "ET"
nevents = 0
OPEN ques$ FOR INPUT AS #11
DO UNTIL EOF(11)
LINE INPUT #11, record$
questiontype$ = MID$(record$, 3, 2)
badchar$ = MID$(questiontype$, 1, 1)
IF badchar$ = "C" OR badchar$ = "c" OR badchar$ = "R" OR badchar$ = "r" THEN
      MID$(questiontype$, 1, 1) = " "
END IF
num = VAL(questiontype$)
IF num <> 0 THEN
      nevents = nevents + 1
      observedevent$(nevents) = eventstr(num + 1)
END IF
tboret$ = MID$(record$, 3, 2)
IF tboret$ = " X" THEN
      nevents = nevents + 1
      observedevent$(nevents) = eventstr(1)
END IF
IF tboret$ = "XX" THEN
      nevents = nevents + 1
      observedevent$(nevents) = eventstr(12)
END IF
LOOP
checkevents:
CLS
PRINT nevents; " events found in question file."
FOR i = 1 TO nevents
PRINT observedevent$(i); " ";
NEXT i
PRINT
correctorder = 1
FOR i = 1 TO 12
IF observedevent$(i) <> eventstr(i) THEN
      PRINT "This is not a standard event order."
```

```
        correctorder = 0
        GOTO outofloop
END IF
NEXT i
outofloop:
IF correctorder = 1 THEN
        PRINT "This set of questions follows the standard order."
END IF
        numcontrols = 0
        numrelevants = 0
        FOR i = 1 TO nevents
        IF observedevent$(i) = eventstr(5) THEN
                numcontrols = numcontrols + 1
                indexc(1) = i
                PRINT "C1 is observed event # "; i
        END IF
        IF observedevent$(i) = eventstr(6) THEN
                numrelevants = numrelevants + 1
                indexr(1) = i
                PRINT "R1 is observed event # "; i
        END IF
        IF observedevent$(i) = eventstr(7) THEN
                numcontrols = numcontrols + 1
                indexc(2) = i
                PRINT "C2 is observed event # "; i
        END IF
        IF observedevent$(i) = eventstr(8) THEN
                numrelevants = numrelevants + 1
                indexr(2) = i
                PRINT "R2 is observed event # "; i
        END IF
        IF observedevent$(i) = eventstr(10) THEN
                numcontrols = numcontrols + 1
                indexc(3) = i
                PRINT "C3 is observed event # "; i
        END IF
        IF observedevent$(i) = eventstr(11) THEN
                numrelevants = numrelevants + 1
                indexr(3) = i
                PRINT "R3 is observed event # "; i
        END IF
        NEXT i
PRINT "There are "; numcontrols; " controls and "; numrelevants; " relevants."
ncrpair = numcontrols
IF numcontrols > numrelevants THEN ncrpair = numrelevants
```

```
PRINT ncrpair; " control-relevant pairs can be processed."
PRINT "Event order will be based on these."
CALL sort(indexc(), indexc(), ncrpair)
CALL sort(indexr(), indexr(), ncrpair)
REM checkevents:
INPUT "Do you want to modify the event order "; modevent
IF modevent = 1 THEN
    FOR i = 1 TO nevents
    PRINT observedevent$(i);
    INPUT e$
    IF e$ <> "" THEN observedevent$(i) = e$
    NEXT i
    GOTO checkevents
END IF
FOR i = 1 TO nevents
eventstr(i) = observedevent$(i)
NEXT i
xnevents = nevents
FOR jj = 1 TO nevents
defaultloc(jj) = (2 * jj - 1) * (80! / xnevents) / 2!
NEXT jj
PRINT
10 : INPUT "0=whole chart, 1=control-relevant pairs, 2=single responses: "; icase
CLS
IF icase = 2 THEN
PRINT "The observed is (event number is in parentheses):"
FOR i = 1 TO nevents
PRINT eventstr(i); "("; i; ") "
NEXT i
PRINT
    INPUT "Enter the number of a single response to display "; sr
    gmax = 0: gmin = 2 ^ 16 - 1: cmax = 0: cmin = gmin: tmax = 0: tmin = gmin: amax = 0:
amin = gmin
    CLS
    FOR i = zeros(sr) TO zeros(sr + 1) - 1
    GET #1, i, rec
    j = i - zeros(sr) + 1
    gs(j) = rec.gsr
    card(j) = rec.cardio
    thor(j) = rec.thoracic
    abdomin(j) = rec.abdominal□
    gmin = xmin(gmin, gs(j)): gmax = xmax(gmax, gs(j))
    cmin = xmin(cmin, card(j)): cmax = xmax(cmax, card(j))
    tmin = xmin(tmin, thor(j)): tmax = xmax(tmax, thor(j))
    amin = xmin(amin, abdomin(j)): amax = xmax(amax, abdomin(j))
```

```
    NEXT i
    n = zeros(sr + 1) - zeros(sr)
REM  PRINT gmin; gmax
REM  INPUT "Press a key:"; k$
REM     FOR k = 1 TO 20
REM     PRINT GS(k);
REM     NEXT k
REM     INPUT "Press a key:"; k$
    CLS
    SCREEN 12□
    WIDTH 80, 60
    scale = 640
    WINDOW (0, 0)-(n, scale)
    s4 = scale / 5
    FOR i = 2 TO n
    gs1 = 4 * s4 + s4 * (gs(i - 1) - gmin) / (gmax - gmin)
    gs2 = 4 * s4 + s4 * (gs(i) - gmin) / (gmax - gmin)
    card1 = 3 * s4 + s4 * (card(i - 1) - cmin) / (cmax - cmin)
    card2 = 3 * s4 + s4 * (card(i) - cmin) / (cmax - cmin)
    thor1 = 2 * s4 + s4 * (thor(i - 1) - tmin) / (tmax - tmin)
    thor2 = 2 * s4 + s4 * (thor(i) - tmin) / (tmax - tmin)
    abdom1 = s4 + s4 * (abdomin(i - 1) - amin) / (amax - amin)
    abdom2 = s4 + s4 * (abdomin(i) - amin) / (amax - amin)
    LINE (i - 1, gs1)-(i, gs2)
    LINE (i - 1, card1)-(i, card2)
    LINE (i - 1, thor1)-(i, thor2)
    LINE (i - 1, abdom1)-(i, abdom2)
    NEXT i
    LINE (0, 0)-(n, scale), , B
    FOR i = 1 TO 4
    LINE (0, i * s4)-(n, i * s4)
    NEXT i
    FOR i = 1 TO n / 30
    LINE (30 * i, 3 * s4 - scale / 120)-(30 * i, 3 * s4 + scale / 120)
    NEXT i
    LOCATE 52, 40: PRINT "File: "; file$
    LOCATE 56, 40: PRINT score$; " "; confirm$
    LOCATE 54, 40: PRINT "Response "; eventstr(sr)
    LOCATE 11, 2: PRINT "GSR"
    LOCATE 23, 2: PRINT "Cardio"
    LOCATE 35, 2: PRINT "Thoracic"
    LOCATE 47, 2: PRINT "Abdominal"
    LOCATE 52, 2: PRINT "GSR Range        "; gmin; ", "; gmax
    LOCATE 54, 2: PRINT "Cardio Range     "; cmin; ", "; cmax
    LOCATE 56, 2: PRINT "Thoracic Range   "; tmin; ", "; tmax
```

```
    LOCATE 58, 2: PRINT "Abdominal Range "; amin; ", "; amax
    WHILE INKEY$ = ""
    WEND
    CLS
ELSEIF icase = 0 THEN
    CLS
    SCREEN 12
    WIDTH 80, 60
    scale = 640
    s4 = scale / 4.25
    WINDOW (0, 0)-(count, scale)
    FOR i = 1 TO count
    GET #1, i, rec2
    REM c11 = ((rec1.gsr - c1min) / (c1max - c1min)) * s4 + 3 * s4
    c12 = ((rec2.gsr - c1min) / (c1max - c1min)) * s4 + 3 * s4
    REM c21 = ((rec1.cardio - c2min) / (c2max - c2min)) * s4 + 2 * s4
    c22 = ((rec2.cardio - c2min) / (c2max - c2min)) * s4 + 2 * s4
    REM c31 = ((rec1.thoracic - c3min) / (c3max - c3min)) * s4 + s4
    c32 = ((rec2.thoracic - c3min) / (c3max - c3min)) * s4 + s4
    REM c41 = ((rec1.abdominal - c4min) / (c4max - c4min)) * s4
    c42 = ((rec2.abdominal - c4min) / (c4max - c4min)) * s4
    IF rec2.event = 0 THEN LINE (i, 0)-(i, scale), 7
    IF rec2.event = 1 THEN LINE (i, 0)-(i, scale), 9
    IF rec2.event = 2 THEN LINE (i, 0)-(i, scale), 4
    PSET (i, c12)
    PSET (i, c22)
    PSET (i, c32)
    PSET (i, c42)
    REM LINE (2 * i - 1, c11)-(2 * i, c12)
    REM LINE (2 * i - 1, c21)-(2 * i, c22)
    REM LINE (2 * i - 1, c31)-(2 * i, c32)
    REM LINE (2 * i - 1, c41)-(2 * i, c42)
    NEXT i
    LINE (0, 0)-(count, 0)
    LINE (0, s4)-(count, s4)
    LINE (0, 2 * s4)-(count, 2 * s4)
    LINE (0, 3 * s4)-(count, 3 * s4)
    FOR j = 1 TO count / (150) - 1
    LINE (j * 150, 2 * s4 - scale / 120)-(j * 150, 2 * s4 + scale / 120)
    NEXT j
    FOR j = 1 TO nevents
    yloc = 80# * (zeros(j + 1) + zeros(j)) / (2# * count)
    IF zeros(j + 1) = 0 OR zeros(j) = 0 THEN yloc = defaultloc(j)
    REM LOCATE 59, yloc: PRINT eventstr(j);
    LOCATE 29, yloc: PRINT eventstr(j);
```

```
    NEXT j
    LINE (0, 0)-(count, scale), , B
    LOCATE 2, 2: PRINT file$; " "; score$; " "; confirm$
    WHILE INKEY$ = ""
    WEND
ELSEIF icase = 1 THEN
        SCREEN 12
        WIDTH 80, 60
        scale = 640
        s5 = scale / 5
        FOR pairnumber = 1 TO ncrpair
        startc = zeros(indexc(pairnumber))
        finishc = zeros(indexc(pairnumber) + 1)
        startr = zeros(indexr(pairnumber))
        finishr = zeros(indexr(pairnumber) + 1)
        gu = 0: gl = 2 ^ 16 - 1: cu = 0: cl = gl
        tu = 0: tl = gl: au = 0: al = gl
        n = finishc - startc + finishr - startr
        FOR j = 1 TO n
        IF j <= finishc - startc THEN i = startc + j - 1
        IF j > finishc - startc THEN i = startr + (j - finishc + startc - 1)
        GET #1, i, rec
        gs(j) = rec.gsr: card(j) = rec.cardio
        thor(j) = rec.thoracic: abdomin(j) = rec.abdominal
        eventmarker = rec.event
        IF eventmarker = 1 AND j <= finishc - startc THEN aqc = j
        IF eventmarker = 2 AND j <= finishc - startc THEN qc = j
        IF eventmarker = 1 AND j > finishc - startc THEN aqr = j
        IF eventmarker = 2 AND j > finishc - startc THEN qr = j
        gu = xmax(gu, gs(j)): gl = xmin(gl, gs(j))
        cu = xmax(cu, card(j)): cl = xmin(cl, card(j))
        tu = xmax(tu, thor(j)): tl = xmin(tl, thor(j))
        au = xmax(au, abdomin(j)): al = xmin(al, abdomin(j))
        NEXT j
        xnd = finishc - startc + 1
        nd = finishc - startc + 1
        ncontrolsamples = finishc - startc
        nrelevantsamples = finishr - startr
        WINDOW (0, 0)-(n, scale)
        FOR j = 2 TO n
        gs1 = (11 / 3) * s5 + s5 * (gs(j - 1) - gl) / (gu - gl)
        gs2 = (11 / 3) * s5 + s5 * (gs(j) - gl) / (gu - gl)
        card1 = (8 / 3) * s5 + s5 * (card(j - 1) - cl) / (cu - cl)
        card2 = (8 / 3) * s5 + s5 * (card(j) - cl) / (cu - cl)
        thor1 = (5 / 3) * s5 + s5 * (thor(j - 1) - tl) / (tu - tl)
```

```
        thor2 = (5 / 3) * s5 + s5 * (thor(j) - tl) / (tu - tl)
        abdom1 = (2 / 3) * s5 + s5 * (abdomin(j - 1) - al) / (au - al)
        abdom2 = (2 / 3) * s5 + s5 * (abdomin(j) - al) / (au - al)
        LINE (j - 1, gs1)-(j, gs2)
        LINE (j - 1, card1)-(j, card2)
        LINE (j - 1, thor1)-(j, thor2)
        LINE (j - 1, abdom1)-(j, abdom2)
        NEXT j
        LINE (nd, (2 / 3) * s5)-(nd, scale)
        LINE (1, (2 / 3) * s5)-(1, scale)
        FOR i = 1 TO 4
        LINE (0, i * s5 - s5 / 3)-(n, i * s5 - s5 / 3)
        NEXT i
        xsec = n / 30!
        FOR i = 1 TO xsec - 1
        LINE (30 * i, 3 * s5 - s5 / 3 - scale / 120)-(30 * i, 3 * s5 - s5 / 3 + scale / 120)
        NEXT i
        LINE (qc, s5 - s5 / 3)-(qc, scale), , , &H707: LINE (qr, s5 - s5 / 3)-(qr, scale), , ,
&H707
        LINE (aqc, s5 - s5 / 3)-(aqc, scale), , , &H707: LINE (aqr, s5 - s5 / 3)-(aqr, scale), , ,
&H707
        LINE (0, 0)-(n, scale), , B
        LOCATE 2, 2: PRINT "Control"
        LOCATE 2, 80 * xnd / n + 3: PRINT "Relevant"
        LOCATE 60 * (s5 - s5 / 3) / scale, 36: PRINT "GSR"
        LOCATE 60 * (2 * s5 - s5 / 3) / scale, 34: PRINT "Cardio"
        LOCATE 60 * (3 * s5 - s5 / 3) / scale, 32: PRINT "Thoracic"
        LOCATE 60 * (4 * s5 - s5 / 3) / scale, 31: PRINT "Abdominal"
        LOCATE 60 * (4 * s5 + s5 / 3) / scale + 2, 2: PRINT "Chart "; file$; "
Control-Relevant
Pair # "; pairnumber
        LOCATE 60 * (4 * s5 + s5 / 3) / scale + 3, 2: PRINT score$; " "; confirm$
        LOCATE 60 * (4 * s5 + s5 / 3) / scale + 4, 2: PRINT "Control Samples=";
ncontrolsamples; " Relevant Samples="; nrelevantsamples
        LOCATE 60 * (4 * s5 + s5 / 3) / scale + 5, 2: INPUT "Go to next set of C-R pairs?
(1=yes) "; inext
        IF inext <> 1 GOTO nomore
        CLS
        NEXT pairnumber
nomore:
END IF
tryitagain:
CLS
SCREEN 0
option$(2) = "Loop through again"
```

```
option$(3) = "Process another chart or subject"
option$(4) = "Create a database file of control-relevant pairs for the current chart"
option$(1) = "Quit"
PRINT "Scroll through the options using RETURN.  Enter 1 to select."
jj = 0
iselect = 0
DO UNTIL iselect = 1
jj = jj + 1
iopt = jj MOD 4
LOCATE 2, 1: PRINT SPACE$(78)□
LOCATE 2, 1: PRINT option$(iopt + 1);
INPUT iselect
LOOP
IF iopt = 1 THEN GOTO 10
IF iopt = 2 THEN
    CLOSE
    GOTO 1000
END IF
IF iopt = 3 THEN GOTO filegeneration
IF iopt = 0 THEN GOTO termination
filegeneration:
CLS
filer = file$□
MID$(filer, 10, 2) = "RC"
filet = filer
MID$(filet, 10, 1) = "T"
PRINT "Creating raw database file "; filer; " and transformed database file "; filet
PRINT "Please be patient..."
OPEN filer FOR BINARY AS #5
OPEN filet FOR BINARY AS #6
iguilt = VAL(MID$(score$, 7, 1))
iconfirm = VAL(MID$(confirm$, 9, 1))
REM ncrpair = 3
ichan = 5
PUT #5, , filer: PUT #6, , filet
PUT #5, , iguilt: PUT #6, , iguilt
PUT #5, , iconfirm: PUT #6, , iconfirm
PUT #5, , ncrpair: PUT #6, , ncrpair
PUT #5, , ichan: PUT #6, , ichan
FOR icrpair = 1 TO ncrpair
PRINT "Transforming C-R pair # "; icrpair
startc = zeros(indexc(icrpair))
finishc = zeros(indexc(icrpair) + 1)
startr = zeros(indexr(icrpair))
finishr = zeros(indexr(icrpair) + 1)
```

```
ncontrolsamples = finishc - startc
nrelevantsamples = finishr - startr
nsamples = finishc - startc + finishr - startr
PUT #5, , ncontrolsamples: PUT #6, , ncontrolsamples
PUT #5, , nrelevantsamples: PUT #6, , nrelevantsamples
FOR j = 1 TO nsamples
IF j <= finishc - startc THEN i = startc + j - 1
IF j > finishc - startc THEN i = startr + (j - finishc + startc - 1)
GET #1, i, rec
gs(j) = rec.gsr: card(j) = rec.cardio
thor(j) = rec.thoracic: abdomin(j) = rec.abdominal
event(j) = rec.event
PUT #5, , rec
NEXT j
PRINT "Transforming GSR"
CALL robust(gs(), gst(), nsamples, gmed, gdevmed)
PRINT "Transforming Cardio"
CALL robust(card(), cardt(), nsamples, cmed, cdevmed)
PRINT "Transforming Thoracic"
CALL robust(thor(), thort(), nsamples, tmed, tdevmed)
PRINT "Transforming Abdominal"
CALL robust(abdomin(), abdomint(), nsamples, amde, adevmed)
PRINT "Done"
FOR j = 1 TO nsamples
recf.gsr = gst(j): recf.cardio = cardt(j)
recf.thoracic = thort(j): recf.abdominal = abdomint(j)
recf.event = event(j)
PUT #6, , recf
NEXT j
NEXT icrpair
PRINT "Finished with C-R pairs"
OPEN "C_R_HIST.txt" FOR APPEND AS #44
PRINT #44, "C-R pairs extracted from "; filerec$; " on "; DATE$; " "; TIME$
CLOSE 44
GOTO tryitagain
termination:
CLOSE
KILL "question.tmp"
KILL "files.dir"
KILL "charts.dir"
END

DEFSNG I-N
SUB robust (arrayin(), arrayout(), n AS INTEGER, arraymed, devmed)
DIM dev(n)
```

```
STATIC dev
CALL sort(arrayin(), arrayout(), n)
IF n MOD 2 <> 0 THEN
      xmed = arrayout((n + 1) / 2)
ELSE
      xmed = .5 * (arrayout(n / 2) + arrayout(n / 2 + 1))
END IF
FOR i = 1 TO n
dev(i) = ABS(arrayout(i) - xmed)
NEXT i
CALL sort(dev(), arrayout(), n)
IF n MOD 2 <> 0 THEN
      dmed = arrayout((n + 1) / 2)
ELSE
      dmed = .5 * (arrayout(n / 2) + arrayout(n / 2 + 1))
END IF
FOR i = 1 TO n
arrayout(i) = (arrayin(i) - xmed) / dmed
NEXT i
arraymed = xmed: devmed = dmed
END SUB

SUB sort (arrayin(), arrayout(), n AS INTEGER)
DIM ra(n)
STATIC ra
FOR i = 1 TO n
ra(i) = arrayin(i)
NEXT i
IF n MOD 2 = 0 THEN
      l = n / 2 + 1
ELSE
      l = FIX(n / 2) + 1
END IF
ir = n
100 :
IF l > 1 THEN
      l = l - 1
      rra = ra(l)
ELSE
      rra = ra(ir)
      ra(ir) = ra(1)
      ir = ir - 1
      IF ir = 1 THEN
            ra(1) = rra
            GOTO 99
```

```
        END IF
END IF
i = l
j = l + l
DO WHILE j <= ir
        IF j < ir THEN
                IF ra(j) < ra(j + 1) THEN j = j + 1
        END IF
        IF rra < ra(j) THEN
                ra(i) = ra(j)
                i = j
                j = j + j
        ELSE
                j = ir + 1
        END IF
ra(i) = rra
LOOP
GOTO 100
99 :
FOR i = 1 TO n
arrayout(i) = ra(i)
NEXT i
END SUB

FUNCTION xmax (x, y)
IF x < y THEN
        xmax = y
    ELSE
        xmax = x
END IF
END FUNCTION

FUNCTION xmin (x, y)
IF x < y THEN
        xmin = x
    ELSE
        xmin = y
END IF
END FUNCTION
```

```
'    VIEWER.BAS
'
'    This program reads the C-R pair files created by DATA2.BAS and
'    displays the control and relevant signals side by side, along with
'     other pertinent information.
'
DECLARE FUNCTION xmin! (x!, y!)
DECLARE FUNCTION xmax! (x!, y!)
DEFINT I-N
TYPE recpiece                'This form used for reading data from a
     gsr AS INTEGER            '.RC* file (raw integer data).
     cardio AS INTEGER
     thoracic AS INTEGER
     abdominal AS INTEGER
     event AS INTEGER
END TYPE
TYPE floatrecpiece             'This form used for reading data from a
     gsr AS SINGLE             '.TC* file (transformed data).
     cardio AS SINGLE
     thoracic AS SINGLE
     abdominal AS SINGLE
     event AS INTEGER
END TYPE
DIM rec AS recpiece, recf AS floatrecpiece
                        'rec is used for raw data, recf for transformed
                        'data.
DIM filename AS STRING * 12
DIM gs(2500), card(2500), thor(2500)
DIM abdomin(2500), event(2500)  AS INTEGER
CLS
SHELL "dir > files.dir"
OPEN "files.dir" FOR INPUT AS #1
OPEN "filelist.dir" FOR OUTPUT AS #2
DO UNTIL EOF(1)
LINE INPUT #1, record$
ext$ = MID$(record$, 10, 2)
IF ext$ = "RC" OR ext$ = "TC" THEN
     filename$ = MID$(record$, 1, 12)
     MID$(filename$, 9, 1) = "."
     PRINT #2, filename$
END IF
LOOP
CLOSE
beginning:
CLS
```

```
PRINT "Select a file to view (return scrolls, 1 selects)"
tryagain:
OPEN "filelist.dir" FOR INPUT AS #1
iselect = 0
DO UNTIL iselect = 1
IF EOF(1) THEN
     CLOSE #1
     GOTO tryagain
END IF
LINE INPUT #1, filerec$
LOCATE 2, 1: PRINT filerec$
INPUT iselect
LOOP
infile$ = filerec$
PRINT infile$; " selected."
OPEN infile$ FOR BINARY AS #2
'***************************************************************************
GET #2, , filename                'File name, string * 12.
GET #2, , iguilt                 'Integer, 0=not guilt, >0=guilty
GET #2, , iconfirm                'Integer, 1=confirmed, 0=not conf.
GET #2, , icrpair                'Integer, # C-R pairs (usually 3).
GET #2, , ichan                  'Integer, # Channels (usually 5).
'***************************************************************************
file$ = filename
score$ = "Guilt=" + STR$(iguilt)
confirm$ = "Confirm=" + STR$(iconfirm)
'***************************************************************************
FOR pairnumber = 1 TO icrpair
GET #2, , ncontrolsamples             'Integer, number of control samples.
GET #2, , nrelevantsamples            'Integer, number of relevant samples.
nsamples = ncontrolsamples + nrelevantsamples
'
FOR j = 1 TO nsamples
IF MID$(file$, 10, 1) = "R" THEN        '"R" designates raw data.
     GET #2, , rec                'Uses integer form of TYPE.
     gs(j) = rec.gsr: card(j) = rec.cardio
     thor(j) = rec.thoracic: abdomin(j) = rec.abdominal
     event(j) = rec.event
ELSE
     GET #2, , recf               'Uses floated form of TYPE, for
                                 'transformed data.
     gs(j) = recf.gsr: card(j) = recf.cardio
     thor(j) = recf.thoracic: abdomin(j) = recf.abdominal
     event(j) = recf.event
END IF
```

```
IF event(j) = 1 AND j <= ncontrolsamples THEN aqc = j
IF event(j) = 2 AND j <= ncontrolsamples THEN qc = j
IF event(j) = 1 AND j > ncontrolsamples THEN aqr = j
IF event(j) = 2 AND j > ncontrolsamples THEN qr = j
NEXT j
'*****************************************************************************
CLS
SCREEN 12
WIDTH 80, 60
scale = 640
s5 = scale / 5
start = 1
middle = ncontrolsamples + 1
finish = nsamples
gu = 0: gl = 2 ^ 16 - 1: cu = 0: cl = gl
tu = 0: tl = gl: au = 0: al = gl
FOR j = start TO finish
gu = xmax(gu, gs(j)): gl = xmin(gl, gs(j))
cu = xmax(cu, card(j)): cl = xmin(cl, card(j))
tu = xmax(tu, thor(j)): tl = xmin(tl, thor(j))
au = xmax(au, abdomin(j)): al = xmin(al, abdomin(j))
NEXT j
n = finish - start + 1
nd = middle - start + 1
xnd = middle - start + 1
qc = qc - start + 1
qr = qr - start + 1
aqc = aqc - start + 1
aqr = aqr - start + 1
WINDOW (0, 0)-(n, scale)
FOR j = start + 1 TO finish
gs1 = (11 / 3) * s5 + s5 * (gs(j - 1) - gl) / (gu - gl)
gs2 = (11 / 3) * s5 + s5 * (gs(j) - gl) / (gu - gl)
card1 = (8 / 3) * s5 + s5 * (card(j - 1) - cl) / (cu - cl)
card2 = (8 / 3) * s5 + s5 * (card(j) - cl) / (cu - cl)
thor1 = (5 / 3) * s5 + s5 * (thor(j - 1) - tl) / (tu - tl)
thor2 = (5 / 3) * s5 + s5 * (thor(j) - tl) / (tu - tl)
abdom1 = (2 / 3) * s5 + s5 * (abdomin(j - 1) - al) / (au - al)
abdom2 = (2 / 3) * s5 + s5 * (abdomin(j) - al) / (au - al)
LINE (j - 1, gs1)-(j, gs2)
LINE (j - 1, card1)-(j, card2)
LINE (j - 1, thor1)-(j, thor2)
LINE (j - 1, abdom1)-(j, abdom2)
NEXT j
LINE (nd, (2 / 3) * s5)-(nd, scale)
```

```
LINE (1, (2 / 3) * s5)-(1, scale)
FOR i = 1 TO 4
LINE (0, i * s5 - s5 / 3)-(n, i * s5 - s5 / 3)
NEXT i
xsec = n / 30!
FOR i = 1 TO xsec - 1
LINE (30 * i, 3 * s5 - s5 / 3 - scale / 120)-(30 * i, 3 * s5 - s5 / 3 + scale / 120)
NEXT i
LINE (qc, s5 - s5 / 3)-(qc, scale), , , &H707: LINE (qr, s5 - s5 / 3)-(qr, scale), , , &H707
LINE (aqc, s5 - s5 / 3)-(aqc, scale), , , &H707: LINE (aqr, s5 - s5 / 3)-(aqr, scale), , , &H707
LINE (0, 0)-(n, scale), , B
LOCATE 2, 2: PRINT "Control"
LOCATE 2, 80 * xnd / n + 3: PRINT "Relevant"
LOCATE 60 * (s5 - s5 / 3) / scale, 36: PRINT "GSR"
LOCATE 60 * (2 * s5 - s5 / 3) / scale, 34: PRINT "Cardio"
LOCATE 60 * (3 * s5 - s5 / 3) / scale, 32: PRINT "Thoracic"
LOCATE 60 * (4 * s5 - s5 / 3) / scale, 31: PRINT "Abdominal"
LOCATE 60 * (4 * s5 + s5 / 3) / scale + 2, 2: PRINT "Chart "; file$; " Control-Relevant Pair # ";
pairnumber
LOCATE 60 * (4 * s5 + s5 / 3) / scale + 3, 2: PRINT score$; " "; confirm$
LOCATE 60 * (4 * s5 + s5 / 3) / scale + 4, 2: PRINT "Control Samples="; ncontrolsamples; "
Relevant Samples="; nrelevantsamples
LOCATE 60 * (4 * s5 + s5 / 3) / scale + 5, 2: PRINT "# C-R Pairs="; icrpair; " # Channels=";
ichan
LOCATE 60 * (4 * s5 + s5 / 3) / scale + 6, 2: PRINT "Press a key to next set of C-R pairs."
WHILE INKEY$ = ""
WEND
CLS
INPUT "Want to write data to an ASCII file (1=yes)"; iwrite
IF iwrite = 1 THEN
    outfile$ = MID$(filename$, 1, 8)
    cr$ = LTRIM$(RTRIM$(STR$(pairnumber)))
    OPEN outfile$ + ".cn" + cr$ FOR OUTPUT AS #6
    OPEN outfile$ + ".rl" + cr$ FOR OUTPUT AS #7
    FOR j = 1 TO ncontrolsamples
    PRINT #6, j; ","; gs(j); ","; card(j); ","; thor(j); ","; abdomin(j)
    NEXT j
    FOR j = ncontrolsamples + 1 TO nsamples
    PRINT #7, j - ncontrolsamples; ","; gs(j); ","; card(j); ","; thor(j); ","; abdomin(j)
    NEXT j
END IF
NEXT pairnumber
CLS
SCREEN 0
INPUT "Want to view another file (1=yes)"; ianother
```

```
IF ianother = 1 THEN
      CLOSE
      GOTO beginning
END IF
CLOSE
KILL "files.dir"
KILL "filelist.dir"
END

FUNCTION xmax (x, y)
IF x < y THEN
      xmax = y
  ELSE
      xmax = x
END IF
END FUNCTION

FUNCTION xmin (x, y)
IF x < y THEN
      xmin = x
  ELSE
      xmin = y
END IF
END FUNCTION
```

# II. Processing of Polygraph Data

## II.1 Introduction

Section II describes our development and evaluation of a data representation technique which will enable evaluation of Artificial Neural Network (ANN) approaches to analysis and classification of multi-dimensional time-varying polygraph signals as an aid to expert examiners. The overall study and polygraph database are described in Section I.

## II.2 Approach

The principal advantage of neural networks resides in their ability to utilize features which are implicitly embedded in the data, not explicitly defined or calculated. This enables the neural network to use only those features which are "necessary and sufficient" for optimal classification. Real-world data, however, rarely exists in a form which is directly mappable to a neural network. Typically, it must be pre-processed in some manner prior to presentation to the neural network.

Proper pre-processing and data representation are the most critical elements in the development of neural network techniques for any application. Our experience indicates that the development of an "optimal" representation requires a combination of insight into the characteristics of the data, an understanding of required performance level (including speed and accuracy ) of the processing, and an understanding of implementation considerations as key components in developing and engineering a neural network solution to a problem such as polygraph classification. Therefore, our overall approach to the development of a data representation technique for use in ultimate processing of polygraph data by neural network includes:

1) Analysis and understanding of polygraph signal characteristics to aid in identifying classes of potential robust data representation techniques which will retain all "necessary and sufficient" information in the resultant representation.
2) Development of data representation and pre-processing techniques requiring minimal explicit definition and/or selection of features, and having minimal impact on distortion and/or elimination of features important to accurate classification.
3) Experimental analysis of data representation techniques, resulting in determination of effectiveness in separating deceptive/non-deceptive subjects.

These steps form the principal focus of this section of the study. Actual processing of polygraph data using neural networks is the focus of a second ongoing study (*Design and Training of an Artificial Neural Network for Polygraph Signal Processing*, Contract #N00014-93-C-0207).

## II.3 Polygraph Data

The polygraph database available for use in this study is described in Section I. Briefly, the raw polygraph data is de-archived, debugged, identified, formatted, organized, and median-normalized (all described in Section I) prior to use in any of the processing described in this section. The database is summarized in Table II-1, and consists of the following:

- 56 subjects
- 41 confirmed deceptives
- 15 confirmed non-deceptives
- Total 436 CR-pairs, ranging from 3 per subject to 9 per subject
- 106 of the Control/Relevant (CR)-pairs are <u>non-deceptive</u>
- 330 of the Control/Relevant (CR)-pairs are <u>deceptive</u>

There are several characteristics of this database which impact potential processing via neural network, including its overall size and the number of deceptives and non-deceptives.

| Subject | Type | CR Pairs | Subject | Type | CR Pairs | Subject | Type | CR Pairs | Subject | Type | CR Pairs |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Deceptive | 9 | 15 | Deceptive | 9 | 29 | Deceptive | 6 | 43 | Deceptive | 9 |
| 2 | Truthful | 3 | 16 | Deceptive | 3 | 30 | Deceptive | 6 | 44 | Truthful | 3 |
| 3 | Deceptive | 6 | 17 | Truthful | 9 | 31 | Deceptive | 9 | 45 | Deceptive | 9 |
| 4 | Deceptive | 9 | 18 | Deceptive | 9 | 32 | Deceptive | 9 | 46 | Truthful | 9 |
| 5 | Truthful | 6 | 19 | Deceptive | 6 | 33 | Deceptive | 9 | 47 | Deceptive | 9 |
| 6 | Truthful | 9 | 20 | Deceptive | 9 | 34 | Deceptive | 6 | 48 | Deceptive | 6 |
| 7 | Truthful | 9 | 21 | Truthful | 6 | 35 | Deceptive | 9 | 49 | Deceptive | 9 |
| 8 | Deceptive | 9 | 22 | Deceptive | 6 | 36 | Deceptive | 9 | 50 | Truthful | 9 |
| 9 | Deceptive | 9 | 23 | Truthful | 3 | 37 | Deceptive | 9 | 51 | Deceptive | 6 |
| 10 | Deceptive | 9 | 24 | Deceptive | 6 | 38 | Deceptive | 9 | 52 | Deceptive | 9 |
| 11 | Truthful | 9 | 25 | Deceptive | 9 | 39 | Deceptive | 9 | 53 | Truthful | 7 |
| 12 | Deceptive | 9 | 26 | Deceptive | 9 | 40 | Deceptive | 9 | 54 | Deceptive | 9 |
| 13 | Truthful | 9 | 27 | Truthful | 6 | 41 | Truthful | 9 | 55 | Deceptive | 9 |
| 14 | Deceptive | 9 | 28 | Deceptive | 6 | 42 | Deceptive | 9 | 56 | Deceptive | 6 |

poly.7

Table II-1. Polygraph database characteristics.

The database itself is small in terms of the number of subjects. In order to train a neural network properly, a sufficient number of representative training examples must be available. Given the range of variability which characterizes polygraph data, 56 subjects may be insufficient, unless it is homogeneously spread over the entire classification space.

There are nearly three times more deceptives than non-deceptives in the database. In order to properly train a neural network, a fair representation of both classes must be available. If it could be shown that the 15 non-deceptives were highly representative of the classification space, and that they were tightly clustered (corresponding to minimal variability), then we could fairly train with few examples. However, it is not clear that this is true in this case. Therefore, to be fair, we would need to subdivide the database into training and testing sets by selecting, say, 10 deceptives and 10 non-deceptives for training, with the remainder for testing (5 non-deceptives and 31 deceptives).

In light of the high variability and extremely high dimensionality of the classification space, 10 training examples is unlikely to be sufficient for conclusive demonstration of the effectiveness of neural network processing. In general, extremely high dimensionality data requires a correspondingly high number of training examples for a neural network to learn the space sufficiently to generalize and perform well. However, we can address the effectiveness of a given data representation technique by analyzing how it reduces the size of the classification space without sacrificing the class separability inherent in the raw data. Assuming that the data representation technique is effective, we can estimate bounds on the potential effectiveness of post-processing via neural network or other classification processing.

## II.4 Processing Overview & Preliminary Explorations

The overall approach to the development of a data representation technique, as a pre-cursor to processing by a nonlinear classification technique such as a neural network, involves a number of steps, including:

Selection of training examples. Homogeneous coverage of classification space must be provided in order to ensure optimal performance of the pattern classification processing. This is a system issue which is ultimately dependent upon the feature space used by the pattern classification (ANN) technique. The intent is to provide a representative set of examples which will enable the trained processor to generalize and correctly classify new examples, which may lie anywhere in the space.

Signal normalization. In order to treat all signals equitably, the signals from each polygraph channel are normalized relative to each other. In our signal normalization processing we treat each of the four primary signals (GSR, Cardio, Upper-Respiratory, Lower-Respiratory) independently. The median-transform processing technique described in Section I is applied to each CR-pair prior to any other processing described in this section. This effectively normalizes all signals to a floating-point range of approximately -10 to +10, and allows comparison of CR-pairs relative to each other, and across charts and subjects, by placing all data into a consistent processing range.

Data representation processing. Direct application of a conventional ANN to polygraph signals is unwieldy at best. A conventional ANN is ill equipped to handle the high dimensionality of the equivalent feature vector represented by a 4 channel stream of sampled polygraph signals. To address some of the issues which emerged during our preliminary investigation of the direct application of the ANN, a *Cellular Automaton (CA)* processing approach was developed to process the phaseplot representation of individual polygraph signals. A number of issues prompted this development.

A digitizing (digital sampling) rate of 30 Hz and a typical control (or relevant) question response of approximately 24 seconds combines to yield a total signal length of approximately 720 samples. Since both the control and relevant question response must be presented to an ANN simultaneously in order for it to determine differences and consequent truth/deception, this signal length corresponds to an effective processing signal-length of 2x720 = 1440 samples per signal channel. For 4 polygraph channels, this equates to 4x1440 = 5760 samples to be processed by an ANN for a single CR-pair. Given that we slide a window over each channel signal and gather classification results along the way, we must segment the 1440 samples for each of the four signals into a number of windows. Since this window length must contain enough of the signal to enable the ANN to properly classify the window, we divide the 1440 samples into no more than, say, 16 windows, corresponding to 90 samples per window for each signal. This yields a total input to the ANN, for each window, of 4x90 = 360 samples -- a lot for both the ANN and a standard PC/486 workstation to process. This factor is the same regardless of whether all 4 signals are being presented to the same ANN or to 4 separate ANN's (one for each signal channel). While 360 samples by itself is not prohibitive for an ANN to process - given sufficient processing power - and sub-sampling by a factor of 2 might be used to help, this issue helped to provide an initial impetus to search for potential alternate processing schemes.

A second issue involves the sliding of an ANN along the signal data and gathering classification results along the way. This yields, say, 16 (or 64) decisions from an ANN(s) for a given CR-pair. This raises several further issues. First, since the ANN sees only a portion of the signal at a time (and assuming that the ANN does not contain any temporal encoding) its classification performance is limited by its incomplete view, as would that of an expert examiner placed in the same position. In addition, the combining of results from processing of each window poses a problem of weighting their relative importance. Should the weighting be equal, or time-dependent relative to the beginning of the signal, or should another ANN be trained to determine an optimal weighting?

The third issue follows from the second. In training a conventional ANN, how does it accommodate differences in phase among multiple (four) channels, and across CR-pairs, charts, and subjects? Given a sufficient number of examples covering the classification space, including a homogenous distribution of combinations of phase differences among all of these elements, a conventional ANN could theoretically learn eventually to handle arbitrary signal sets having arbitrary phase relationships. However, the scope of this study and the limited polygraph data available to us precludes performance of the required level of extensive training.

While these issues do not rule out the use of conventional ANN's in processing the polygraph data, they did provide an impetus to explore development of a novel phaseplot representation/CA processing technique which appears to address all of these issues in a satisfactory manner. The phaseplot/CA technique may be characterized as follows:

- The use of the phaseplot is based on the hypothesis that the multi-dimensional data represented by a given CR-pair identifies the presence of an attractor in phase space which corresponds to deception or non-deception in the generator of the data (the subject).
- The CA, which is a fine-grained locally-interconnected massively parallel processing plane, handles the entire signal simultaneously for each channel of each CR-pair without requiring windowing and its corresponding problems as described above for a conventional ANN.
- The CA handles the multi-dimensional CR-pair data in phase-space thus eliminating differences in phase between channels, and across CR-pairs, charts, and subjects.
- By training in phase-space the CA also effectively eliminates the issues of data scaling across all examples. The hypothesis is that the phase-space for the polygraph data is self-similar, in that the attractors and corresponding multi-dimensional phase-space trajectories for deception and non-deception correspond to a given attractor - dependent only upon the source of the data (the subject's source of deception/truth) as measured by the polygraph - and independent of the actual scaling of the data.

Decision processing. The ultimate intent of this process is to provide high accuracy decision-assistance to the polygraph examiner. Performance is highly dependent upon the effectiveness of the data representation, processing, and pattern classification techniques employed. This is addressed in more detail in our follow-on study.

## II.5 Software Overview

The overall structure of the software developed for this study is illustrated in Figure II-1. The individual processing elements are described in greater detail below. All software has been prototyped on a PC486/33 system in Visual Basic Pro 3.0 for Windows, and has undergone literally hundreds of revisions as processing algorithms and user-interfaces were developed throughout the study.

Briefly, the polygraph database, (described in Section I), provides the primary source of data to the processing chain. The database consists of multiple data files, each corresponding to a set of two or three CR-pairs for a given subject. There may be more than one CR-pair file per subject, and each file (corresponding to an original polygraph chart) may contain up to three CR-pairs. Each subject may have up to nine CR-pairs.

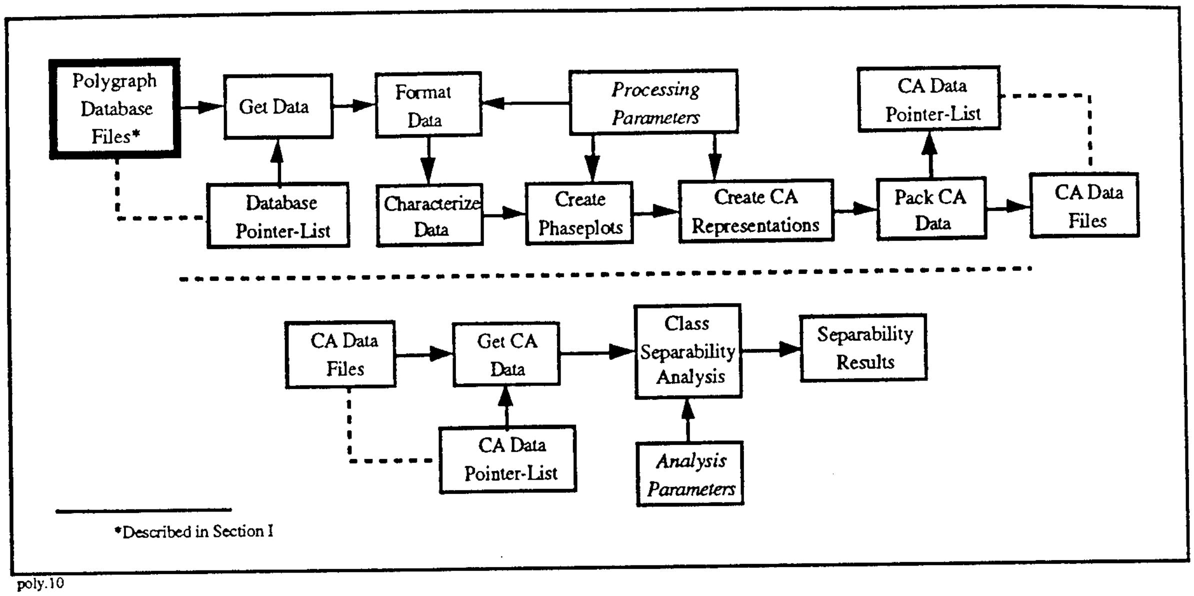



Figure II-1. Software overview.

Individual CR-pairs are accessed by standard interface modules which enable access to arbitrary subjects, CR-pairs, and polygraph channels, using a database pointer list which identifies and locates all of files in the database. The data is then formatted and characterized prior to creation of phaseplots and mapping into cellular automata, as described below. The CA data is then packed to reduce file storage requirements, and stored along with a CA database pointer list as 56 separate files, each consisting of CA data representing up to nine 4-channel CR-pairs. This intermediate storage technique greatly reduces the amount of computational and file-access (I/O) time required in the class separability analysis process (i.e., database I/O, computation of phaseplots, and CA-mappings are performed only once). Finally, highly interactive presentation and analysis software was developed to enable the rapid and insightful analysis of class separability intended to yield key results for this study.

## II.6 Processing Chain

The processing chain developed for analysis of data representation and processing effectiveness consists of four principal elements, as shown in Figure II-2:

- Signal pre-processing
- Data representation and processing
- Computation of distances
- Analysis of class separability

Our data representation and processing approach is uniquely characterized by its strict adherence to a self-imposed guideline that all processing be data-independent. This requirement constrains the envelope of potential solutions to those for which data-dependent features are neither used nor allowed to impact development of the processing approach. This results in extremely efficient processing, since data-driven decisions are completed eliminated and the potential for highly parallel implementations is greatly enhanced. Our approach mimics that of nature - as in the eye's retina, which does not change its operation for each different image presented to it, but does recognize certain features (e.g., edges) in images and pre-processes them in a highly parallel manner before sending both raw and processed information to the brain. This processing is built-in, and is always present and operating, independent of the actual data present. The following subsections discuss the four principal elements of the processing chain in more detail.

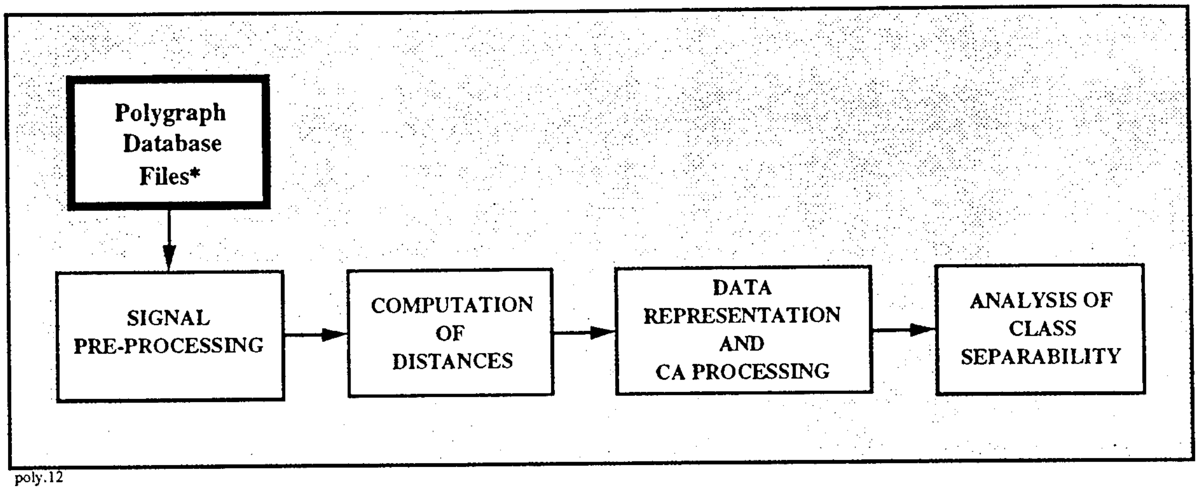



Figure II-2. Overview of processing chain.

### II.6.1 *Signal Pre-Processing*

As shown in Figure II-3, signal pre-processing draws data from the working database on a subject-by-subject basis. For each subject, there are anywhere from 3 to 9 CR-pair files. For each of these CR-pairs there are 4 channels of sampled data (GSR, Cardio, Upper-Respiratory, Lower-Respiratory) corresponding to the *control* question and 4 channels corresponding to the *relevant* question. In this report, this data is referred to interchangeably as "channel data" or "signal data."

The data is handled by the pre-processing on a channel-by-channel basis. For each channel, the initial number of samples ranges from 600 to over 1000 samples. The pre-processing prepares a uniform window of data by limiting the number of samples to 512 for each channel. In addition, the processing removes DC biases in channels 2-4 (Cardio, Upper-Respiratory, Lower-Respiratory) in order to emphasize the time-varying characteristics of these signals and reduce ambiguities. The resulting data is termed "raw" data, as shown.

Finally, for each channel the minimum and maximum of the raw data is computed for each CR-pair. This enables scaling of the control and relevant data, relative to each other, within a fixed amplitude range expected by all subsequent processing. After scaling, the resultant data is termed "pre-processed" data, as shown.

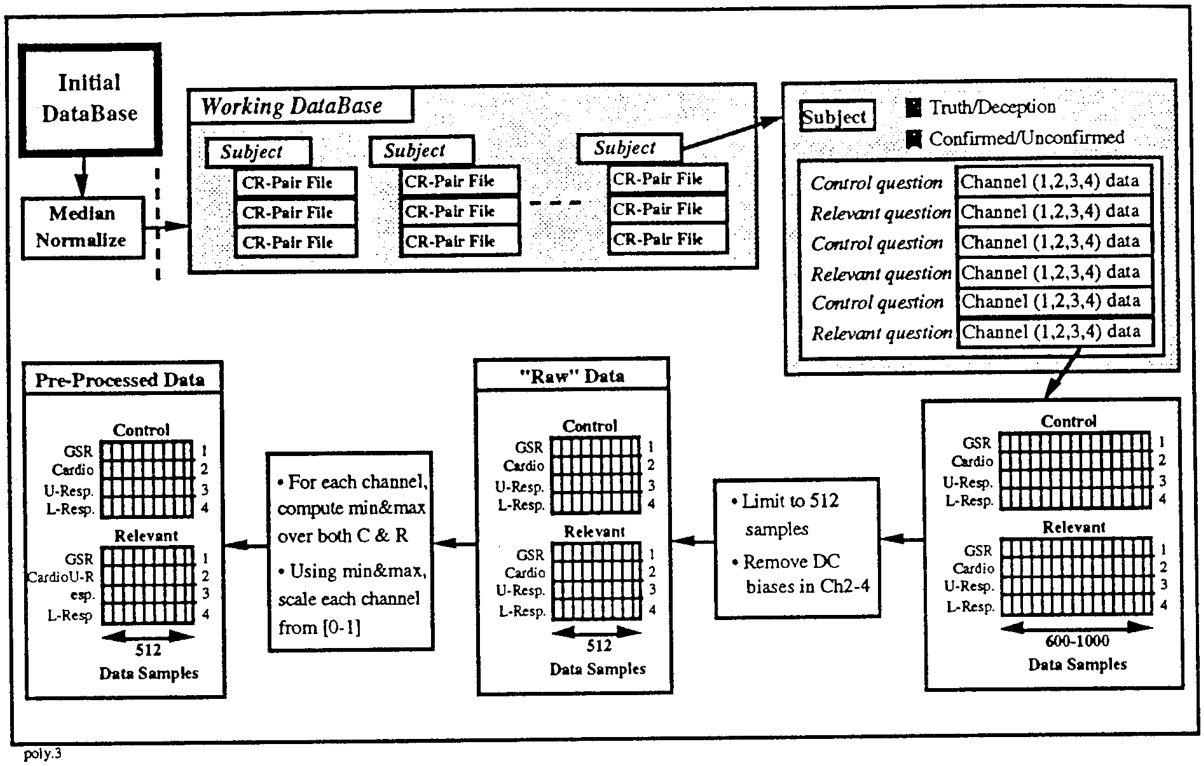



Figure II-3. Signal pre-processing.

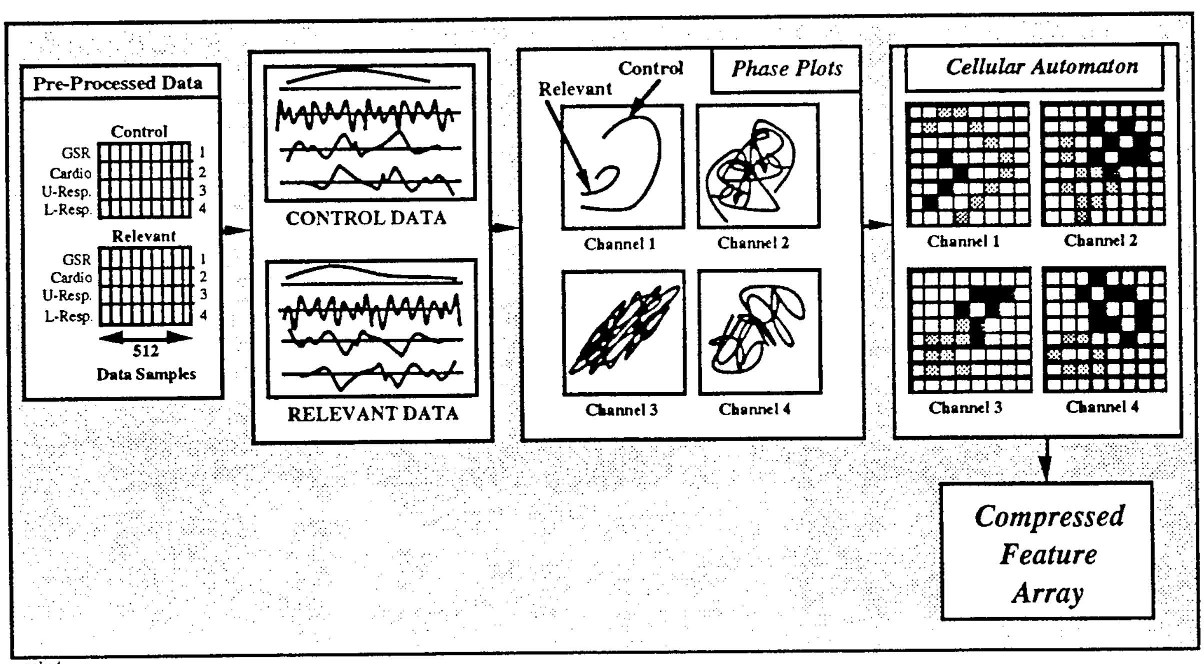
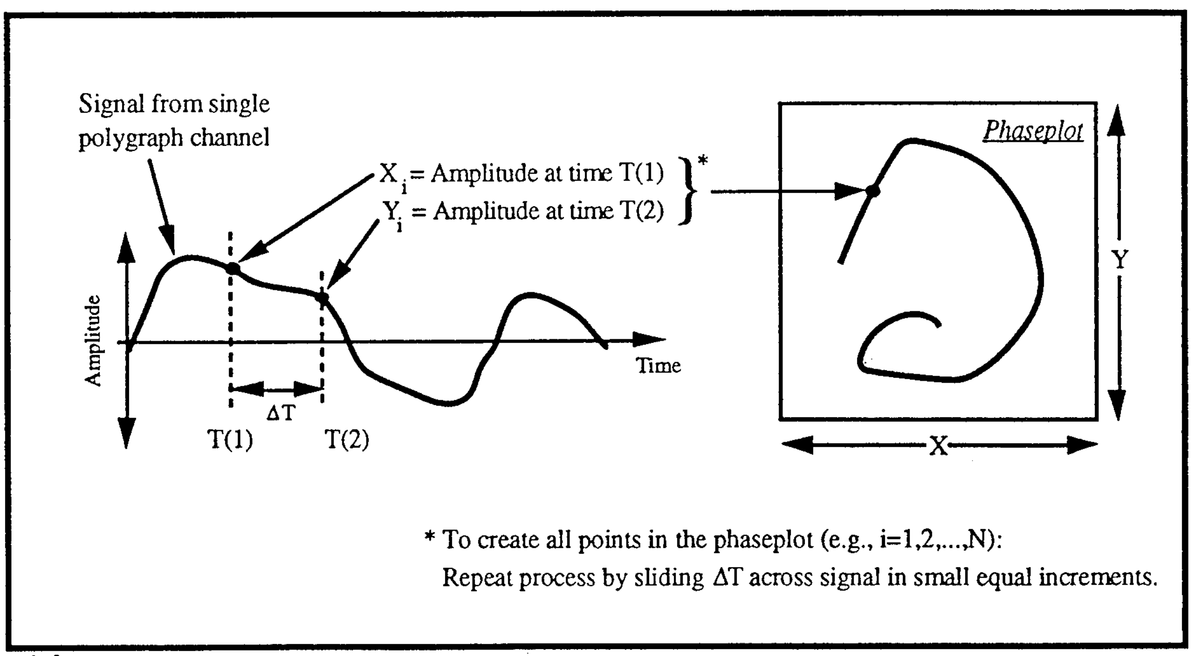



Figure II-4. Data representation and processing.

## II.6.2 *Data Representation and Processing*

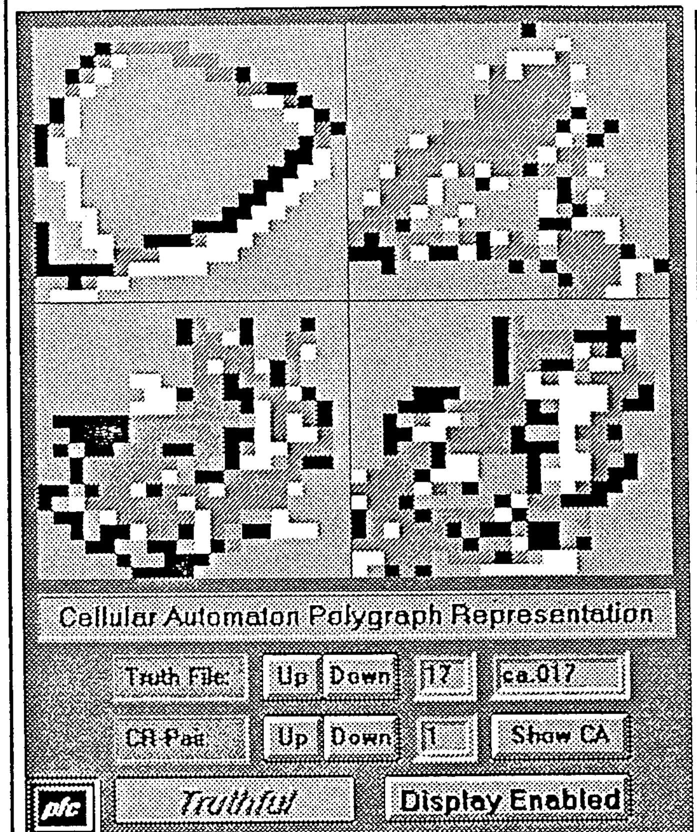
As shown in Figure II-4, data representation and processing draws from the pre-processed data on a CR-pair by CR-pair basis. For each channel, the data representation technique creates a phaseplot using a fixed sample time-delay. As illustrated in the figure and in the images shown in Screen II-1 through Screen II-6 on the following pages, the resultant phaseplots demonstrate a marked difference between the control and the relevant signals (particularly noticeable in the GSR plots). It is this difference in phase space that led us to believe that data representation via phaseplot would accomplish the dual objectives of reducing dimensionality without sacrificing separability and subsequent classifiability.

Figure II-5 illustrates the mapping of a generic signal into a phaseplot representation. For each polygraph channel signal a delay time ($\Delta T$) is determined as an approximate function of the channel's fundamental frequencies. $\Delta T$ may be different for each channel, but is held constant for a given experiment across all subjects and CR-pairs. Pairs of amplitude points, separated by $\Delta T$, are then selected from the signal to yield a single [X,Y] point in the phaseplot plane. The complete phaseplot is created by sliding the $\Delta T$ "window" over the entire signal in small increments (usually defined by the sampling rate of the digitized signal data). One of the most powerful characteristics of the phaseplot representation is that the phaseplot itself is independent of the "starting" and "ending" points for the signal. That is, the phaseplot is independent of the phase of the signal as defined by its initial sample. This proves to be very useful when comparing signals, overcoming the weaknesses and ambiguities characteristic of conventional cross-correlation signal processing techniques.



poly.9

Figure II-5. Mapping of a generic time-amplitude signal into a phaseplot representation.




Each cell has a 00, 01, 10, or 11 value resulting from mapping of phaseplot data into a 20x20 CA, as described in the text:

[00] Light-Grey = No data

[01] Black = Control

[10] White = Relevant

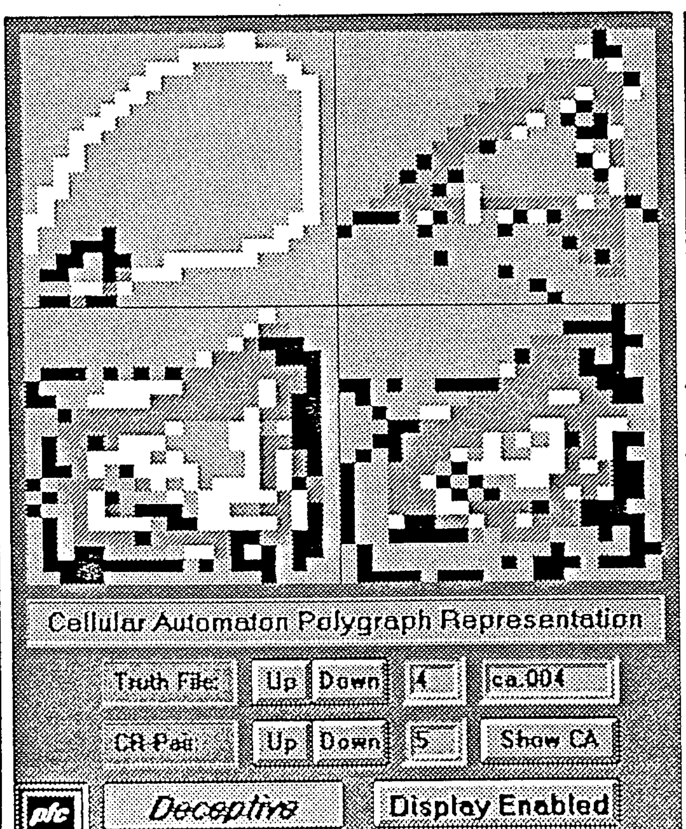
[11] Grey = Ctrl & Relev.

*Display Key*

| GSR | Cardio |
|---|---|
| U-Resp | L-Resp |

- Similarities between ctrl & relev data in GSR phaseplot (similar large "hoops").
- Similarities between ctrl and relev. data in remaining 3 phaseplots. Much overlap, resulting in dark grey.

Screen II-1. Processing output display of phaseplots mapped into four cellular automata for a representative NON-DECEPTIVE polygraph subject.




Each cell has a 00, 01, 10, or 11 value resulting from mapping of phaseplot data into a 20x20 CA, as described in the text:

[00] Light-Grey = No data

[01] Black = Control

[10] White = Relevant

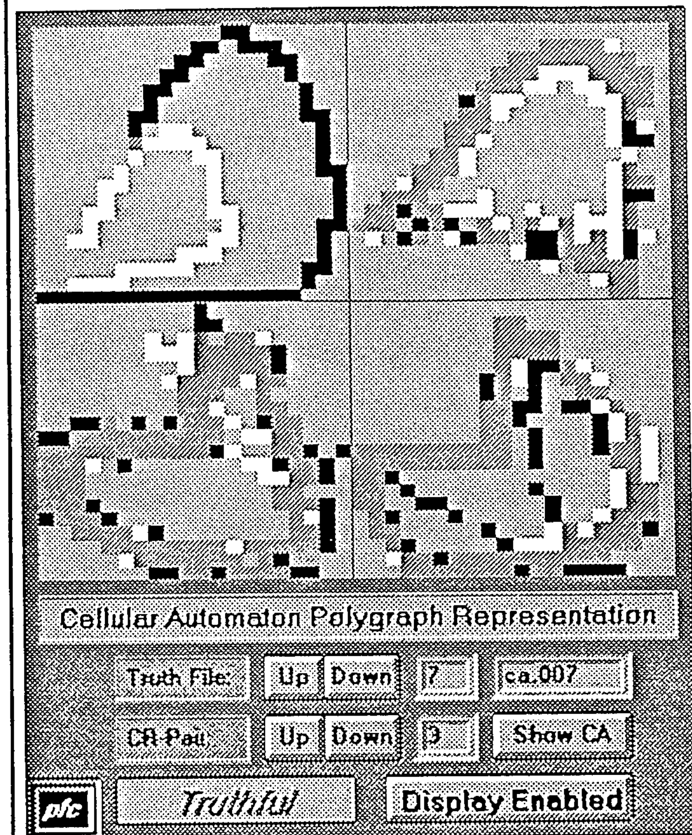
[11] Grey = Ctrl & Relev.

*Display Key*

| GSR | Cardio |
|---|---|
| U-Resp | L-Resp |

- Large "hoop" in relev (white) GSR phaseplot. Indicates strong response to relev. question.
- Large difference between ctrl & relev data in GSR phaseplot.
- Differences between ctrl and relev. data in remaining 3 phaseplots. Relev (black) shows as quite distinct in U-resp &L-resp.

Screen II-2. Processing output display of phaseplots mapped into four cellular automata for a representative DECEPTIVE polygraph subject.




Each cell has a 00, 01, 10, or 11 value resulting from mapping of phaseplot data into a 20x20 CA, as described in the text:

[00] Light-Grey = No data

[01] Black = Control

[10] White = Relevant

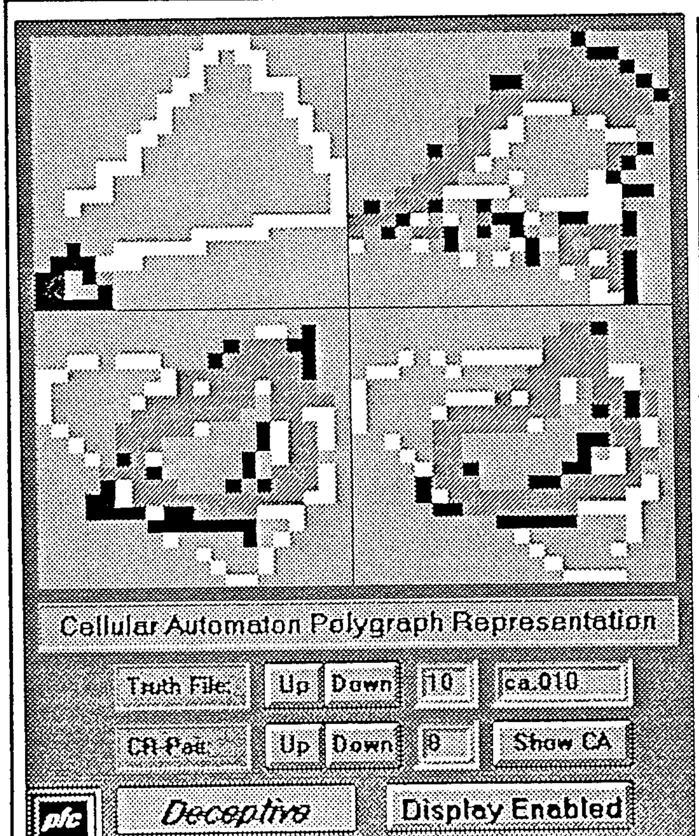
[11] Grey = Ctrl & Relev.

*Display Key*

| GSR | Cardio |
|---|---|
| U-Resp | L-Resp |

- Large "hoop" in control (black) GSR phaseplot. Indicates strong response to control question.
- Similarities between ctrl and relev. data in remaining 3 phaseplots. Much overlap, resulting in dark grey.

Screen II-3. Processing output display of phaseplots mapped into four cellular automata for a representative NON-DECEPTIVE polygraph subject.




Each cell has a 00, 01, 10, or 11 value resulting from mapping of phaseplot data into a 20x20 CA, as described in the text:

[00] Light-Grey = No data

[01] Black = Control

[10] White = Relevant

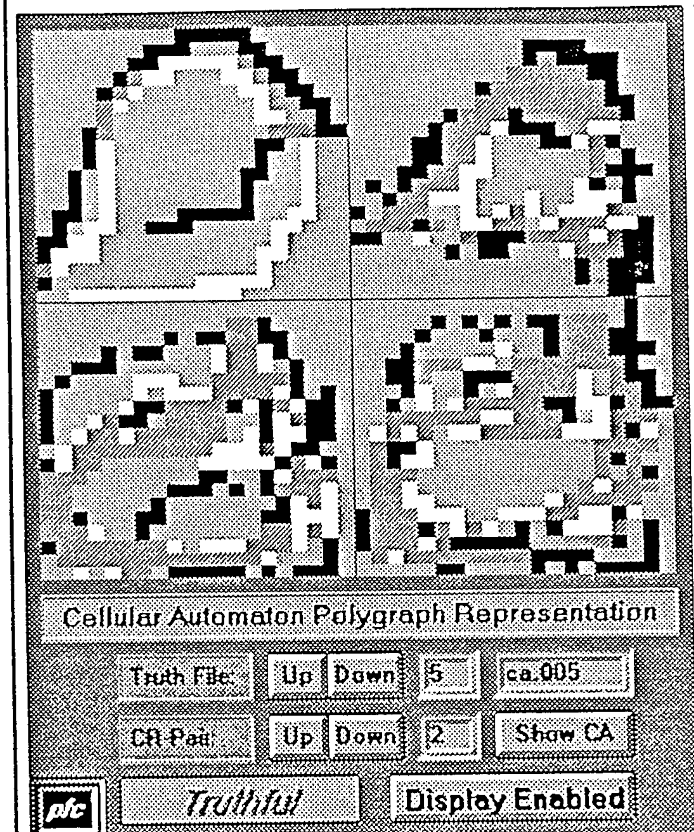
[11] Grey = Ctrl & Relev.

*Display Key*

| GSR | Cardio |
|---|---|
| U-Resp | L-Resp |

- Large "hoop" in relev (white) GSR phaseplot
- Large difference between ctrl & relev data in GSR phaseplot
- Multiple differences between ctrl and relev. in remaining 3 phaseplots. Ctrl (white) shows as quite distinct in U-resp &L-resp.

Screen II-4. Processing output display of phaseplots mapped into four cellular automata for a representative DECEPTIVE polygraph subject.




Each cell has a 00, 01, 10, or 11 value resulting from mapping of phaseplot data into a 20x20 CA, as described in the text:

[00] Light-Grey = No data

[01] Black = Control

[10] White = Relevant

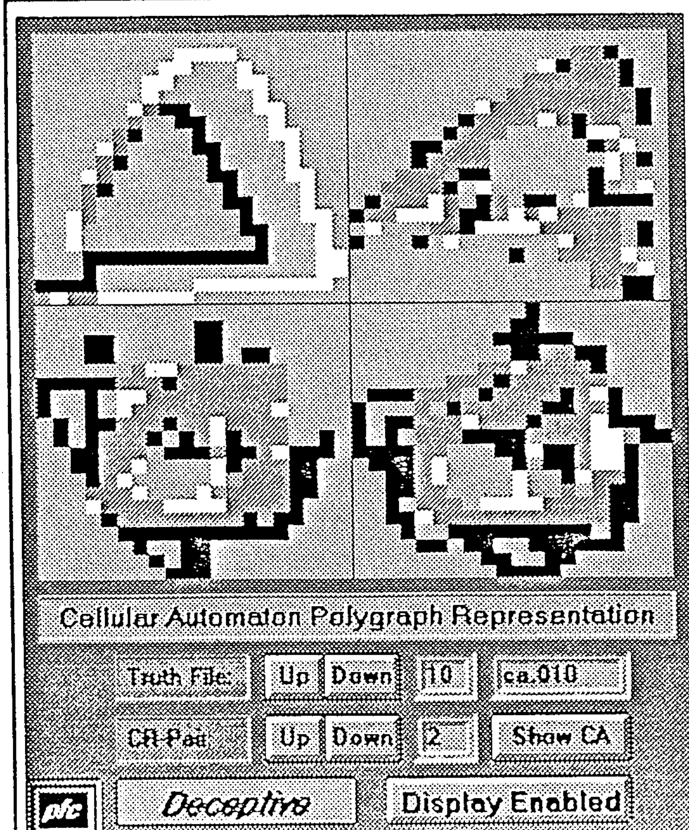
[11] Grey = Ctrl & Relev.

*Display Key*

| | |
|---|---|
| GSR | Cardio |
| U-Resp | L-Resp |

- Similarities between ctrl & relev data in GSR phaseplot (similar large "hoops").
- Similarities between ctrl and relev. data in remaining 3 phaseplots. Much overlap, resulting in dark grey.

Screen II-5. Processing output display of phaseplots mapped into four cellular automata for a representative NON-DECEPTIVE polygraph subject.




Each cell has a 00, 01, 10, or 11 value resulting from mapping of phaseplot data into a 20x20 CA, as described in the text:

[00] Light-Grey = No data

[01] Black = Control

[10] White = Relevant

[11] Grey = Ctrl & Relev.

*Display Key*

| | |
|---|---|
| GSR | Cardio |
| U-Resp | L-Resp |

- Larger "hoop" in relev (white) GSR phaseplot. Indicates strong response to relev. question.
- Some difference between ctrl & relev data in GSR phaseplot.
- Differences between ctrl and relev. data in remaining 3 phaseplots. Relev (black) shows as quite distinct in U-resp &L-resp.

Screen II-6. Processing output display of phaseplots mapped into four cellular automata for a representative DECEPTIVE polygraph subject.

Using the polygraph data, the phaseplot is mapped into a simple cellular automaton (CA) configured essentially multi-planar memory having an interleaved fine-grained processing surface. This mapping accomplishes a number of things. By its very structure, the mapping combines neighboring spatial data to produce a reduced-resolution representation of the phaseplot, corresponding to a lower dimensional feature vector (or more appropriately in the 2-D case of the CA, a *feature array*). This is consistent with the data-independent processing which uniquely characterizes our data representation, processing, and neural network classification approach, as it effectively enhances processing speed while reducing dimensionality without sacrificing important information content. Our goal is to ensure that all processing is independent of specific explicit features of the data. That is, processing should operate completely independently of the actual data (except for scaling and normalization, as noted above).

Data resolution in the CA corresponds to the size (length and width, in cells) of the CA. A larger CA, say 100x100 cells, results in relatively high-resolution encoding of the data. A smaller CA, say 10x10 cells, results in a much lower-resolution representation of phaseplot information, effectively combining local spatial neighborhood data into single cells. Any number of encoding schemes may be used, including representation of the number of neighborhood points included within a given cell in the final CA. We have chosen to simplify the encoding initially, in order to minimize processing time and maximize efficiency in terms of both storage space and processing speed. If more detailed information is ultimately required, we can include more "complicating" features in the model as required to accommodate desired performance goals. We begin with a very "lean" approach.

As shown in multi-planar cellular automaton structure depicted in Figure II-6, each phaseplot contains both control and relevant data for a single channel, and resides in two of five independent planes of the CA. Multiple encoding schemes are possible for each of these planes, with the simplest involving a 1-bit code for each cell in the plane, where for CA planes 1 & 2 [xy]:

[00] = No data present

[01] = Cell "set" by *control* question response data present

[10] = Cell "set" by *relevant* question response data present

[11] = Cells "set" by *control* and *relevant* question response data both present

In this way, a single byte in the prototype software model can contain all four channels of data for a given cell in the CA representing a given CR-pair, greatly reducing memory and file storage requirements for the CA feature array. The third and fourth planes of the CA are used to store a CA feature array for a second CR-pair whose distance from the first is to be computed for separability analysis and/or classification. Finally, the fifth ("middle") CA plane is a computing surface used to determine the distance between the 2 CA feature arrays, as described in the next section.

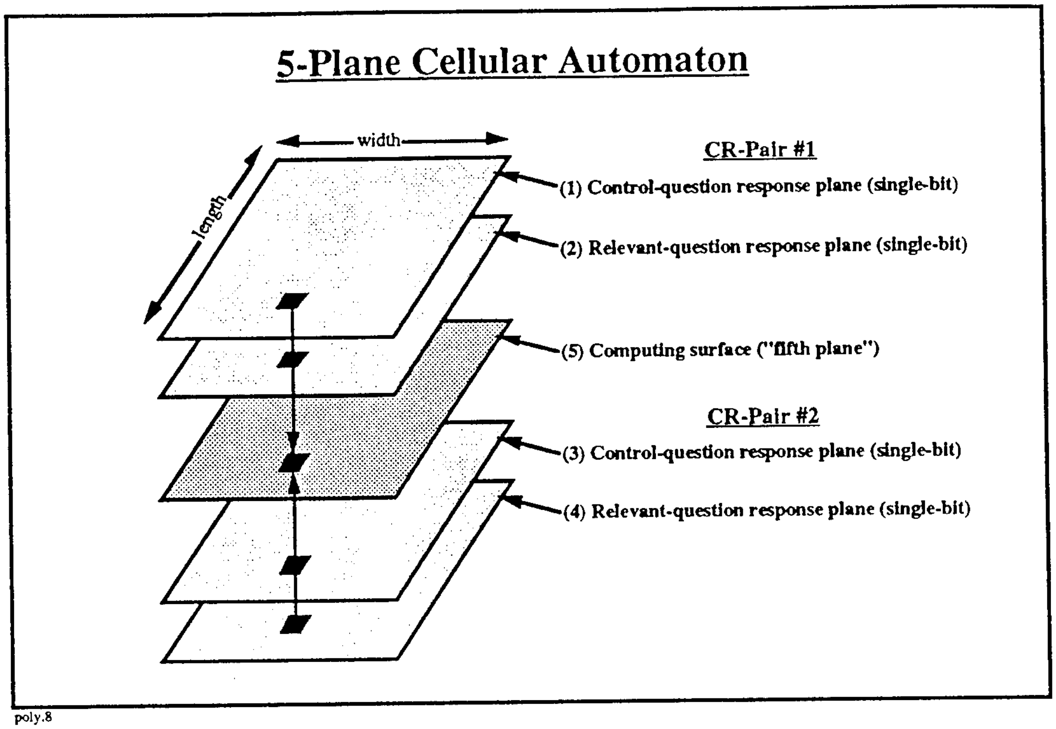



Figure II-6. Multi-planar cellular automaton structure.

The unique computational architecture represented by the CA's multi-planar structure enables a variety of data manipulation and filtering processes to be elegantly embedded in any number of simple but powerful distance computations that might be selected for implementation in the plane. For example, independent spatial spreading of data in each plane may serve to reduce distances which might be due to "near-misses," thereby improving overall performance. This same spreading may also be used to represent a (perhaps weighted) composite of multiple CR-pairs. On the other hand, Laplacian or other two-dimensional filter processing may be used to emphasize higher frequency information contained in edges of certain phaseplots, say for selected channels, resulting in potentially improved classification performance.

A large number of distance measurement techniques are enabled by this unique five-plane CA structure. We have experimented with multiple distance metrics, including the following:

1) Simple cell-to-cell cross-plane differencing, involving cross-plane bit-to-bit comparison and CA-spanning summation operations.
2) Two-dimensional fractal dimension computation of planes 1&2 and 3&4, both in combination and separately, with the measure of distance between CR-pairs corresponding to differences in fractal dimension. We also briefly explored the potential use of the fractal dimension of raw (non-phaseplot) data for each channel.
3) X- and Y-axis "histogramming" of data in each plane, with subsequent differencing among various combinations of resultant "linearized" representations of the CA feature arrays along each axis.

4) Elegant time-domain computation of the near-optimal Hausdorff distance between two planar data sets ("images") by counting the iterations required for spreading images in various combinations of the planes to intersect to a significant level of completeness.
5) Application of these techniques in combination with high-pass filtering, low-pass filtering, and other techniques for enhancing the data contained in planes 1&2 and 3&4.

After prototyping and qualitatively analyzing these techniques and variations thereof, we settled on technique #1 (simple cell-to-cell differencing) for its apparent potential for excellent performance as well as its inherent simplicity and ease of efficient implementation. Relying on appropriate selection of CA size (and corresponding data resolution) to effectively accomplish some low-pass filtering, or smearing of the data, together with this simple distance measure enabled us to accomplish our analysis of class separability without resorting to more complex and computationally-intensive distance metrics.

The 5-plane CA architecture has proven to be extremely versatile for exploring alternative processing approaches. In addition, it can theoretically operate completely in parallel, computing among all "cells" simultaneously and resulting in extremely fast processing of polygraph data (potentially faster than real-time, even on non-parallel machines), for processing of archived data.

After mapping, the resultant data is termed a "compressed feature array," as shown. At this point, the essence of the data has been retained, its dimensionality has been greatly reduced, and it is ready for classification processing by a neural network or other methods..

### II.6.3 *Computation of Distances*

As shown in Figure II-7, analysis of class separability is based on distances computed between compressed feature arrays for all 56 subjects against each of the 15 compressed feature arrays for all non-deceptive subjects. Specifically, for each of the 436 CR-pairs available in the database (for all 56 subjects), the distance to all 106 CR-pairs corresponding to the 15 non-deceptive subjects is computed (using technique #1 described above). This is performed for all four channels, resulting in 436x106x4 = 184,864 distances. Of these, 106x4=424 distances by definition (for identical CR-pairs) are identically zero, resulting in a total of 184,440 usable distances. These distances correspond to all inter-CR-pair distances for all deceptive subjects and all non-identical non-deceptive subjects. This data forms the basis for our estimate of class separability.

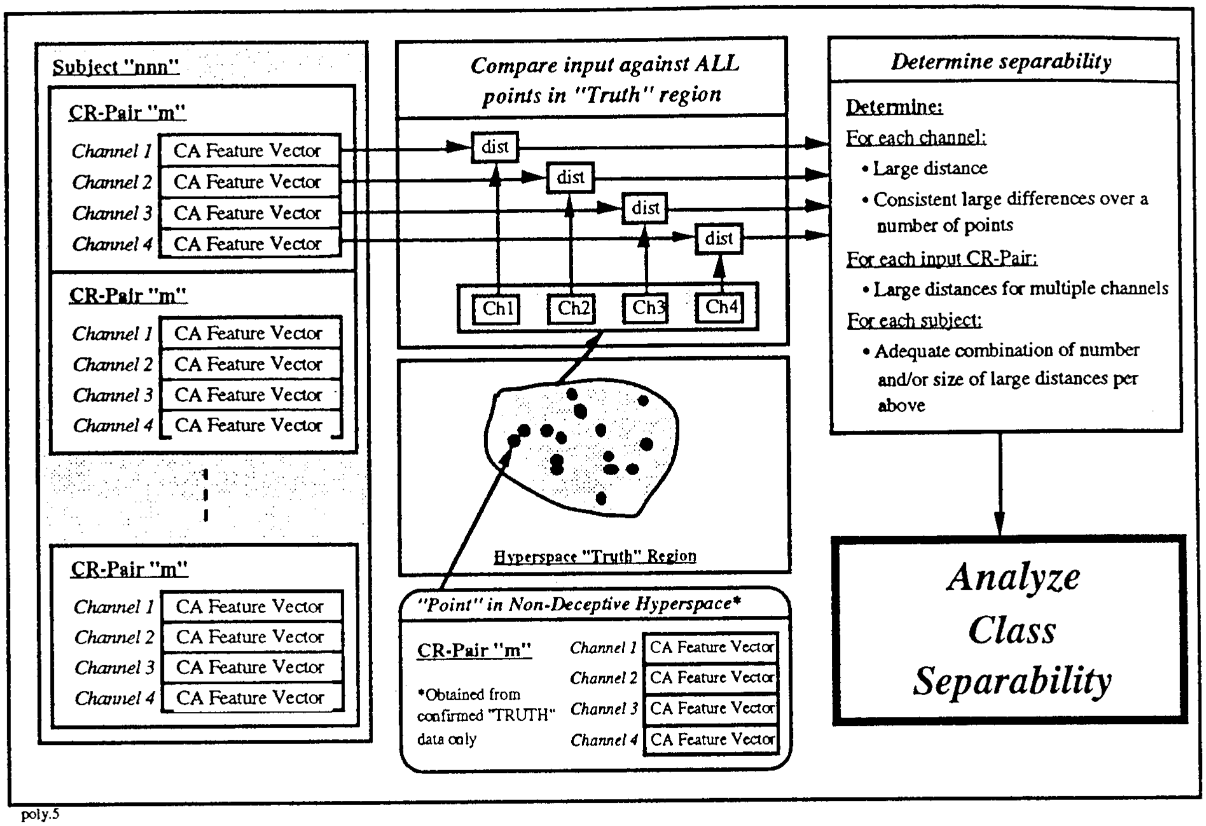



Figure II-7. Computation of distances.

### II.6.4 *Analysis of Class Separability*

The process for analyzing class separability is illustrated in the steps shown in Figure II-8. After initially working with the data and a variety of hierarchical (step-wise) classification schemes, we determined that classification of subjects into deceptive/non-deceptive classes cannot be handled in a hierarchical fashion, as we had originally anticipated. Determination of deception/non-deception at the channel level, followed by a combination to classify at the CR-pair level, followed by a further combination to classify at the subject level, neither follows the expert examiner's implicit approach, nor yields acceptable performance by automatic processing. The decision of the examiner may hinge upon a small number of artifacts in a single channel for a small number of question responses. This corresponds to a highly non-linear process and is not conducive to well-structured step-wise hierarchical classification techniques.

Therefore, our analysis of class separability reflects the inherently nonlinear nature of the polygraph classification process by seeking to identify necessary and sufficient significant differences between the nearest of deceptive and non-deceptive CA feature arrays. Specifically, our approach includes the following:

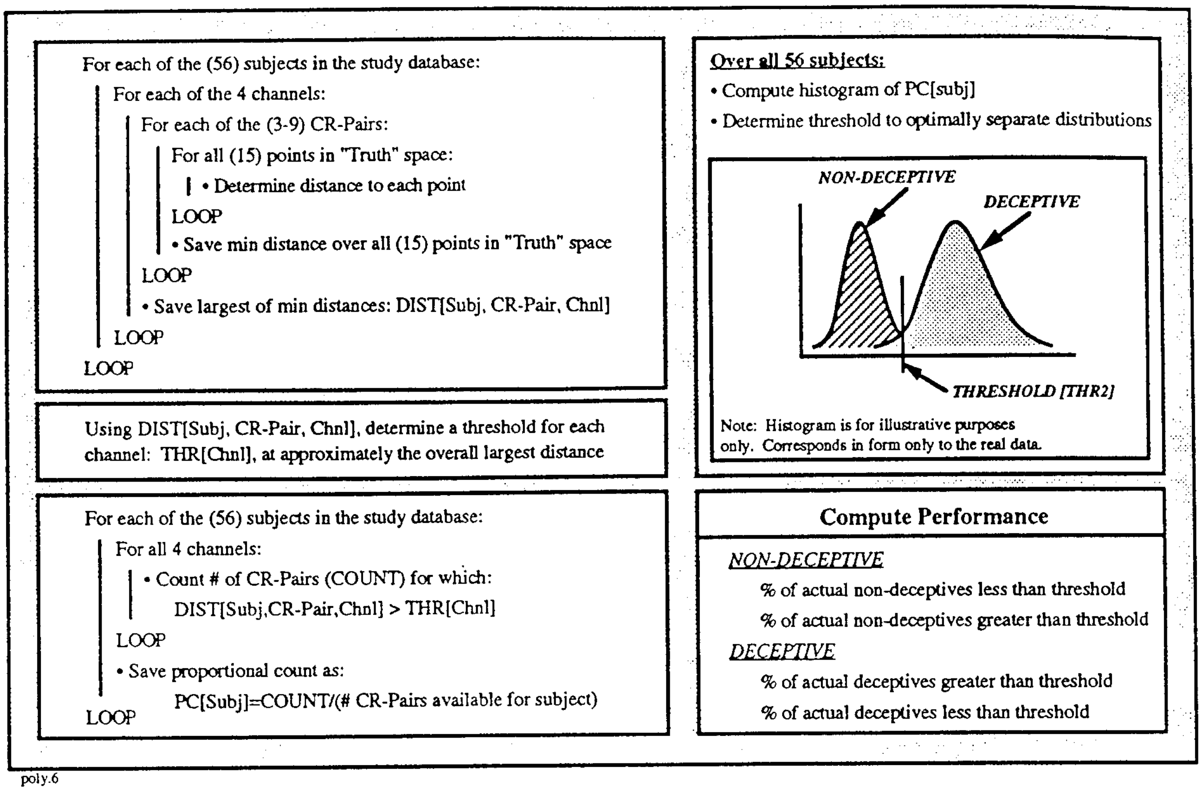



Figure II-8. Analysis of class separability.

- For each subject and channel we determine and save the largest of the smallest distances between all of the subject's CR-pairs and the 106 non-deceptive CR-pairs.
- Then, over all four channels, we count the number of these distances which exceed a given threshold.
- This count is then normalized by the number of available CR-pairs for the given subject, and is associated with the given subject, revealing those potentially few (large) distances that correspond to significant differences between subjects. Deceptive subjects should exhibit more large differences than non-deceptive subjects.
- These differences are then used to determine classifiability. As illustrated in the figure, a histogram of these counts corresponding to all subjects was found to reveal a bimodal structure, as expected, with non-deceptive subjects corresponding to a lower bias than deceptive subjects.
- Finally, appropriate selection of a "classification" threshold using any of a number of classical techniques - minimizing false classifications and maximizing true - results in a quantified analysis of class separability of the data based on this parameter.

## II.7 Results

Summary data used in determining class separability for one of our final experimental runs is tabulated in Table II-2. Four performance values may be calculated from this data:

% of actual non-deceptives less than threshold

% of actual non-deceptives greater than threshold

% of actual deceptives less than threshold

% of actual deceptives greater than threshold

Specifically, this data yielded our most encouraging results:

**87%** **of actual non-deceptives were classified as non-deceptive**

**13% of actual non-deceptives were classified as deceptive**

**95%** **of actual deceptives were classified as deceptive**

**5% of actual deceptives were classified as non-deceptive**

The few misclassifications represented in these results are evenly split: 2 misclassified deceptives (out of 41) and 2 misclassified non-deceptives (out of 15). The two non-deceptives were just slightly over the classification threshold, into the deceptive region of the classification space, and could potentially be called inconclusive. The two deceptives were strongly within the non-deceptive territory of the space, and may be considered at this point to be outliers, requiring further analysis. If in fact they do define an actual deceptive sub-region buried within the non-deceptive region, appropriate (non-linear) neural network classification techniques should help in their classification by effectively "carving out" the sub-region and identifying it as deceptive. Although we could have assigned confidence levels to the classifications and thereby potentially included some inconclusives in our results, we chose instead to focus on strict binary classifiability in order to determine strict performance bounds.

Overall, while these results are very promising, we must keep in mind that they are for a limited set of data: i.e., 56 subjects, of which only 15 were non-deceptive. The techniques developed in this study appear to work very well on this data, but generalization to a claim that they will successfully address the overall polygraph classification problem requires more extensive evaluation and demonstration. That is, a higher confidence could be assigned to our results if we had processed, say, several hundred of each type of subject.

## II.8 Summary

A novel data representation and processing approach has been developed, involving representation of polygraph channel data (CR-pairs) as phaseplots which are subsequently mapped into multi-planar cellular automata (CA) for rapid data-independent processing and computation of (simple but effective) distances in classification space. Analysis of class separability as a function of these techniques coupled with a non-hierarchical classification strategy has yielded 95% correct classification of deceptive subjects and 87% correct classification of non-deceptive subjects. These results represent lower bounds on the potential performance of artificial neural network and/or other classifiers applied to the CA feature arrays which represent the polygraph data.

While these results are promising, further development of post-feature-extraction classifiers and extensive evaluation against a much larger database of confirmed subjects is required in order to demonstrate and validate the true potential for the overall polygraph classification problem. In addition, a number of variations in data representation and processing parameters could be explored in order to verify potential impact on performance, including:

- Varying of $\Delta T$ in phaseplot generation would result in variations in the phaseplot "image" and corresponding differences in class separability potential.
- Direct CA processing of an 8-dimensional phaseplot corresponding to a composite of the four polygraph channels.
- Adaptive determination of data representation and processing parameters, possibly by neural network or genetic algorithm, to assist in optimizing overall performance.

10/25/93 1:33:44 PM

| | | | | |
|---|---|---|---|---|
| 1 001 | 41 34 32 28 27 23 23 20 20 | 28 25 23 23 22 22 22 20 17 | 34 34 33 32 29 28 26 23 22 | 33 32 28 27 25 25 25 23 20 |
| 0 002 | 28 26 16 00 00 00 00 00 00 | 19 16 16 00 00 00 00 00 00 | 23 22 22 00 00 00 00 00 00 | 29 20 20 00 00 00 00 00 00 |
| 1 003 | 34 28 22 16 13 10 00 00 00 | 34 25 23 23 23 11 00 00 00 | 30 30 28 27 26 20 00 00 00 | 39 36 35 34 29 27 00 00 00 |
| 1 004 | 35 32 28 27 27 26 23 20 14 | 20 19 17 17 16 14 13 11 11 | 33 32 30 28 28 28 26 23 22 | 33 33 32 30 30 25 23 23 20 |
| 0 005 | 40 34 32 30 26 17 00 00 00 | 23 17 17 16 14 14 00 00 00 | 28 26 26 23 23 23 00 00 00 | 28 28 28 27 26 22 00 00 00 |
| 0 006 | 42 40 34 33 30 30 29 27 23 | 23 20 16 16 14 14 14 13 13 | 34 28 28 27 26 23 22 20 20 | 28 27 27 26 25 23 23 23 20 |
| 0 007 | 40 35 34 30 28 25 25 23 23 | 14 13 13 11 11 10 10 08 08 | 23 20 20 20 19 19 19 17 17 | 29 20 20 17 17 17 17 14 14 |
| 1 008 | 40 36 29 28 28 28 27 20 19 | 27 26 26 25 23 23 22 22 19 | 34 33 32 27 27 25 23 23 22 | 34 29 29 29 28 26 26 23 23 |
| 1 009 | 36 36 34 29 26 20 17 17 16 | 17 17 17 16 14 14 14 13 11 | 28 27 27 25 25 23 23 22 20 | 29 28 28 28 26 26 25 23 23 |
| 1 010 | 39 36 34 32 28 23 23 20 14 | 17 16 16 13 11 11 11 11 08 | 46 35 34 34 32 32 29 25 20 | 40 39 38 36 33 26 25 22 17 |
| 0 011 | 29 28 28 23 22 22 20 20 17 | 28 17 14 14 14 13 11 08 08 | 30 28 23 23 23 22 20 17 17 | 28 28 26 25 23 23 23 20 20 |
| 1 012 | 40 38 38 34 34 29 28 23 23 | 25 23 23 22 20 20 19 19 17 | 28 27 27 27 25 25 23 20 17 | 32 28 28 26 26 25 23 23 22 |
| 0 013 | 39 34 34 32 32 29 29 28 17 | 39 28 14 14 14 14 13 11 11 | 46 41 32 32 29 26 25 23 23 | 39 34 32 29 28 27 26 25 20 |
| 1 014 | 36 34 34 33 33 29 26 26 23 | 23 20 19 17 17 16 16 14 14 | 30 29 28 28 27 26 26 23 20 | 32 30 27 27 26 25 23 22 19 |
| 1 015 | 35 30 29 27 20 20 16 13 10 | 33 25 23 23 23 22 17 16 14 | 39 34 29 20 20 20 20 20 17 | 40 38 36 34 34 33 32 29 26 |
| 1 016 | 34 27 20 00 00 00 00 00 00 | 20 19 17 00 00 00 00 00 00 | 30 26 23 00 00 00 00 00 00 | 28 23 20 00 00 00 00 00 00 |
| 0 017 | 45 36 33 26 25 22 17 14 14 | 20 19 19 17 17 14 14 13 13 | 23 23 23 23 22 22 22 20 20 | 27 27 26 25 25 25 22 22 20 |
| 1 018 | 36 30 30 29 28 23 22 19 16 | 28 26 26 25 23 20 16 16 16 | 32 30 29 28 28 27 23 23 22 | 28 27 26 23 23 22 19 19 19 |
| 1 019 | 36 34 33 32 22 20 00 00 00 | 45 23 22 20 20 17 00 00 00 | 32 29 23 23 22 20 00 00 00 | 32 32 28 28 23 22 00 00 00 |
| 1 020 | 42 36 35 34 32 32 32 30 28 | 23 23 23 22 22 22 20 17 16 | 35 34 34 34 32 32 30 30 28 | 33 30 29 29 29 28 26 26 25 |
| 0 021 | 34 32 25 25 23 23 00 00 00 | 14 14 10 08 08 05 00 00 00 | 34 28 27 27 22 22 00 00 00 | 30 28 28 27 25 25 00 00 00 |
| 1 022 | 41 40 30 28 25 23 00 00 00 | 23 22 20 19 16 16 00 00 00 | 32 30 30 28 25 25 00 00 00 | 40 33 32 28 26 25 00 00 00 |
| 0 023 | 34 33 33 00 00 00 00 00 00 | 13 11 11 00 00 00 00 00 00 | 34 28 28 00 00 00 00 00 00 | 26 26 26 00 00 00 00 00 00 |
| 1 024 | 46 39 38 33 32 30 00 00 00 | 34 28 27 27 27 25 00 00 00 | 34 28 28 27 26 23 00 00 00 | 33 29 28 27 26 23 00 00 00 |
| 1 025 | 48 42 40 36 32 30 26 25 23 | 39 23 20 20 20 19 19 19 19 | 35 34 33 33 32 30 30 28 27 | 40 40 34 33 32 32 30 30 27 |
| 1 026 | 40 38 35 34 34 33 32 28 28 | 33 33 33 30 29 23 22 20 20 | 42 32 29 29 28 28 28 28 28 | 52 34 33 28 28 28 28 26 23 |
| 0 027 | 36 32 32 28 27 14 00 00 00 | 17 11 11 07 07 05 00 00 00 | 30 27 23 23 23 23 00 00 00 | 34 29 29 29 27 27 00 00 00 |
| 1 028 | 42 41 40 38 35 35 00 00 00 | 38 32 28 28 28 19 00 00 00 | 46 41 41 40 39 30 00 00 00 | 42 36 34 32 32 28 00 00 00 |
| 1 029 | 54 45 45 42 38 33 00 00 00 | 17 17 17 17 16 16 00 00 00 | 33 28 26 23 23 17 00 00 00 | 30 28 27 23 23 22 00 00 00 |
| 1 030 | 41 40 35 30 27 23 00 00 00 | 17 17 14 13 11 10 00 00 00 | 40 28 28 26 23 22 00 00 00 | 46 30 26 23 22 22 00 00 00 |
| 1 031 | 41 35 34 32 27 25 23 22 20 | 22 19 19 17 17 17 16 14 11 | 32 28 28 28 26 26 25 23 20 | 36 36 30 28 28 27 26 25 23 |
| 1 032 | 38 35 32 29 29 28 28 28 23 | 36 35 32 29 25 23 23 19 19 | 36 34 33 32 32 29 27 26 25 | 40 39 38 34 34 32 29 29 29 |
| 1 033 | 41 40 40 39 34 34 29 27 22 | 32 20 20 20 19 17 17 17 17 | 39 38 30 30 29 29 28 28 27 | 40 35 34 28 28 28 27 26 25 |
| 1 034 | 36 33 33 29 27 26 00 00 00 | 19 19 17 16 14 14 00 00 00 | 30 30 28 27 23 22 00 00 00 | 35 34 28 28 28 27 00 00 00 |
| 1 035 | 52 45 45 44 41 36 35 28 22 | 27 22 19 17 17 17 16 11 11 | 28 28 26 25 25 23 23 23 22 | 36 34 33 28 28 27 25 22 20 |
| 1 036 | 29 28 27 26 26 26 23 23 22 | 40 36 27 23 22 20 20 17 17 | 44 36 35 35 34 34 34 34 34 | 38 38 38 35 35 34 33 32 27 |
| 1 037 | 44 40 36 35 34 33 28 26 25 | 20 19 17 17 17 14 14 14 13 | 32 29 28 28 28 25 25 23 23 | 35 30 29 28 28 26 23 23 20 |
| 1 038 | 41 35 34 30 29 28 27 25 19 | 20 19 17 16 14 14 14 14 11 | 40 28 26 23 23 23 23 23 23 | 42 34 28 28 26 26 23 23 17 |
| 1 039 | 40 40 28 28 27 22 20 20 17 | 34 32 28 28 26 26 23 20 17 | 34 28 28 26 23 23 23 23 22 | 40 40 34 33 30 30 26 25 23 |
| 1 040 | 39 34 34 27 25 22 22 17 16 | 25 23 20 20 20 20 20 17 16 | 30 28 26 25 23 23 22 22 22 | 30 29 28 28 28 26 25 22 22 |
| 0 041 | 40 36 36 36 35 34 33 28 28 | 20 16 14 14 13 13 10 10 10 | 33 30 29 29 29 27 26 25 25 | 30 30 28 28 27 25 23 23 23 |
| 1 042 | 34 29 29 28 28 28 26 23 20 | 41 34 22 20 16 14 13 13 11 | 39 29 25 22 20 20 20 20 17 | 46 35 32 28 28 28 28 28 27 |
| 1 043 | 42 38 35 34 33 33 28 28 26 | 26 20 19 19 17 16 16 16 14 | 35 33 33 32 30 28 27 27 25 | 28 28 28 28 27 26 20 20 19 |
| 0 044 | 34 27 14 00 00 00 00 00 00 | 17 14 14 00 00 00 00 00 00 | 41 29 23 00 00 00 00 00 00 | 28 23 16 00 00 00 00 00 00 |
| 1 045 | 34 30 30 27 26 25 20 20 16 | 26 25 23 22 19 19 19 16 13 | 34 34 30 29 28 28 25 23 22 | 39 38 29 29 28 28 28 25 23 |
| 0 046 | 38 36 35 28 28 28 25 22 16 | 14 14 14 14 14 13 11 07 07 | 27 23 23 23 23 23 23 17 17 | 28 28 27 25 23 23 19 19 16 |
| 1 047 | 40 35 33 28 26 26 25 23 20 | 36 28 26 22 20 20 14 14 14 | 34 28 28 28 27 25 25 23 20 | 36 29 28 28 27 27 25 23 23 |
| 1 048 | 33 28 28 28 25 11 00 00 00 | 28 20 19 19 19 17 00 00 00 | 33 30 28 28 27 23 00 00 00 | 29 26 25 23 23 20 00 00 00 |
| 1 049 | 33 30 27 26 23 20 19 17 11 | 23 22 20 20 19 17 16 16 14 | 33 30 28 27 27 22 22 19 19 | 30 28 26 25 23 23 22 20 20 |
| 0 050 | 41 36 35 32 29 28 28 28 14 | 25 13 13 13 11 11 11 11 11 | 23 22 22 20 20 20 17 16 16 | 20 17 17 17 17 16 16 14 14 |
| 1 051 | 41 40 39 32 30 17 00 00 00 | 19 19 17 16 16 14 00 00 00 | 33 32 28 27 23 20 00 00 00 | 44 36 36 32 28 20 00 00 00 |
| 1 052 | 32 28 28 28 23 22 20 19 16 | 30 20 20 20 17 17 14 14 11 | 28 28 28 28 27 26 25 23 23 | 27 27 26 26 25 23 23 23 17 |
| 0 053 | 33 33 32 32 29 28 23 00 00 | 42 17 17 17 17 16 16 00 00 | 28 27 26 23 23 23 23 00 00 | 27 25 25 25 22 22 22 00 00 |
| 1 054 | 34 28 28 27 23 22 22 19 17 | 32 23 19 17 16 14 14 14 11 | 40 38 34 34 33 28 27 26 20 | 34 32 30 28 28 28 26 23 22 |
| 1 055 | 44 34 33 28 23 20 19 19 19 | 17 17 17 16 16 14 13 11 07 | 44 34 33 33 30 28 28 26 23 | 32 29 29 27 26 23 23 23 14 |
| 1 056 | 35 33 28 22 14 14 00 00 00 | 40 39 34 34 17 14 00 00 00 | 40 40 38 34 26 25 00 00 00 | 40 40 40 38 36 33 00 00 00 |

Table II-2. Processing results summary data used in class separability analysis.

| | | CHNL 1 | CHNL 2 | CHNL 3 | CHNL 4 | | |
|---|---|---|---|---|---|---|---|
| 0 000 | 0 | 45 40 35 30 25 20 15 10 | 45 40 35 30 25 20 15 10 | 45 40 35 30 25 20 15 10 | 45 40 35 30 25 20 15 10 | 8 35 | 20 30 30 |
| 1 001 | 9 | 00 01 01 03 05 09 09 09 | 00 00 00 00 02 08 09 09 | 00 00 00 04 07 09 09 09 | 00 00 00 02 07 09 09 09 | 15 | 0.417 |
| 0 002 | 3 | 00 00 00 00 02 02 03 03 | 00 00 00 00 00 00 03 03 | 00 00 00 00 00 03 03 03 | 00 00 00 00 01 03 03 03 | 00 | 0.000 |
| 1 003 | 6 | 00 00 00 01 02 03 04 06 | 00 00 00 01 02 05 05 06 | 00 00 00 02 05 06 06 06 | 00 00 03 04 06 06 06 06 | 11 | 0.458 |
| 1 004 | 9 | 00 00 01 02 06 08 08 09 | 00 00 00 00 00 01 05 09 | 00 00 00 03 07 09 09 09 | 00 00 00 05 06 09 09 09 | 10 | 0.278 |
| 0 005 | 6 | 00 01 01 04 05 05 06 06 | 00 00 00 00 00 01 04 06 | 00 00 00 00 03 06 06 06 | 00 00 00 00 05 06 06 06 | 02 | 0.083 |
| 0 006 | 9 | 00 02 02 06 08 09 09 09 | 00 00 00 00 00 02 04 09 | 00 00 00 01 05 09 09 09 | 00 00 00 00 05 09 09 09 | 05 | 0.139 |
| 0 007 | 9 | 00 01 02 04 07 09 09 09 | 00 00 00 00 00 00 00 07 | 00 00 00 00 00 04 09 09 | 00 00 00 00 01 03 07 09 | 02 | 0.056 |
| 1 008 | 9 | 00 01 02 02 07 08 09 09 | 00 00 00 00 04 08 09 09 | 00 00 00 03 06 09 09 09 | 00 00 00 01 07 09 09 09 | 14 | 0.389 |
| 1 009 | 9 | 00 00 02 03 05 06 09 09 | 00 00 00 00 00 00 04 09 | 00 00 00 00 05 09 09 09 | 00 00 00 00 07 09 09 09 | 02 | 0.056 |
| 1 010 | 9 | 00 00 02 04 05 08 08 09 | 00 00 00 00 00 00 03 08 | 01 01 02 06 08 09 09 09 | 00 01 04 05 07 08 09 09 | 13 | 0.361 |
| 0 011 | 9 | 00 00 00 00 03 08 09 09 | 00 00 00 00 01 01 02 07 | 00 00 00 01 02 07 09 09 | 00 00 00 00 04 09 09 09 | 02 | 0.056 |
| 1 012 | 9 | 00 01 03 05 07 09 09 09 | 00 00 00 00 01 06 09 09 | 00 00 00 00 06 08 09 09 | 00 00 00 01 06 09 09 09 | 10 | 0.278 |
| 0 013 | 9 | 00 00 01 05 08 08 09 09 | 00 00 01 01 02 02 02 09 | 01 02 02 04 07 09 09 09 | 00 00 01 03 08 09 09 09 | 10 | 0.278 |
| 1 014 | 9 | 00 00 01 05 08 09 09 09 | 00 00 00 00 00 02 07 09 | 00 00 00 01 07 09 09 09 | 00 00 00 02 06 08 09 09 | 06 | 0.167 |
| 1 015 | 9 | 00 00 01 02 04 06 07 09 | 00 00 00 01 02 06 08 09 | 00 00 01 02 03 08 09 09 | 00 01 03 07 09 09 09 09 | 16 | 0.444 |
| 1 016 | 3 | 00 00 00 01 02 03 03 03 | 00 00 00 00 00 01 03 03 | 00 00 00 01 02 03 03 03 | 00 00 00 00 01 03 03 03 | 02 | 0.167 |
| 0 017 | 9 | 01 01 02 03 05 06 07 09 | 00 00 00 00 00 01 05 09 | 00 00 00 00 00 09 09 09 | 00 00 00 00 06 09 09 09 | 03 | 0.083 |
| 1 018 | 9 | 00 00 01 03 05 07 09 09 | 00 00 00 00 04 06 09 09 | 00 00 00 02 06 09 09 09 | 00 00 00 00 03 06 09 09 | 09 | 0.250 |
| 1 019 | 6 | 00 00 01 04 04 06 06 06 | 01 01 01 01 01 05 06 06 | 00 00 00 01 02 06 06 06 | 00 00 00 02 04 06 06 06 | 09 | 0.375 |
| 1 020 | 9 | 00 01 03 08 09 09 09 09 | 00 00 00 00 00 07 09 09 | 00 00 01 08 09 09 09 09 | 00 00 00 02 09 09 09 09 | 20 | 0.556 |
| 0 021 | 6 | 00 00 00 02 04 06 06 06 | 00 00 00 00 00 00 00 03 | 00 00 00 01 04 06 06 06 | 00 00 00 01 06 06 06 06 | 02 | 0.083 |
| 1 022 | 6 | 00 02 02 03 05 06 06 06 | 00 00 00 00 00 03 06 06 | 00 00 00 03 06 06 06 06 | 00 01 01 03 06 06 06 06 | 11 | 0.458 |
| 0 023 | 3 | 00 00 00 03 03 03 03 03 | 00 00 00 00 00 00 00 03 | 00 00 00 01 03 03 03 03 | 00 00 00 00 03 03 03 03 | 01 | 0.083 |
| 1 024 | 6 | 01 01 03 06 06 06 06 06 | 00 00 00 01 06 06 06 06 | 00 00 00 01 05 06 06 06 | 00 00 00 01 05 06 06 06 | 11 | 0.458 |
| 1 025 | 9 | 01 03 04 06 08 09 09 09 | 00 00 01 01 01 05 09 09 | 00 00 01 07 09 09 09 09 | 00 02 02 08 09 09 09 09 | 24 | 0.667 |
| 1 026 | 9 | 00 01 03 07 09 09 09 09 | 00 00 00 04 05 09 09 09 | 00 01 01 02 09 09 09 09 | 01 01 01 03 08 09 09 09 | 17 | 0.472 |
| NON-DECEP. 0 027 | 6 | 00 00 01 03 05 05 05 06 | 00 00 00 00 00 00 01 03 | 00 00 00 01 02 06 06 06 | 00 00 00 01 06 06 06 06 | 03 | 0.125 |
| DECEP. 1 028 | 6 | 00 03 06 06 06 06 06 06 | 00 00 01 02 05 05 06 06 | 01 04 05 06 06 06 06 06 | 00 01 02 05 06 06 06 06 | 22 | 0.917 |
| 1 029 | 6 | 03 04 05 06 06 06 06 06 | 00 00 00 00 00 00 06 06 | 00 00 00 01 03 05 06 06 | 00 00 00 01 03 06 06 06 | 07 | 0.292 |
| 1 030 | 6 | 00 02 03 04 05 06 06 06 | 00 00 00 00 00 00 02 06 | 00 01 01 01 04 06 06 06 | 01 01 01 02 03 06 06 06 | 06 | 0.250 |
| 1 031 | 9 | 00 01 02 04 06 09 09 09 | 00 00 00 00 00 01 07 09 | 00 00 00 01 07 09 09 09 | 00 00 02 03 08 09 09 09 | 07 | 0.194 |
| 1 032 | 9 | 00 00 02 03 08 09 09 09 | 00 00 02 03 05 07 09 09 | 00 00 01 05 09 09 09 09 | 00 01 03 06 09 09 09 09 | 20 | 0.556 |
| 1 033 | 9 | 00 03 04 06 08 09 09 09 | 00 00 00 01 01 04 09 09 | 00 00 02 04 09 09 09 09 | 00 01 02 03 09 09 09 09 | 15 | 0.417 |
| 1 034 | 6 | 00 00 01 03 06 06 06 06 | 00 00 00 00 00 00 04 06 | 00 00 00 02 04 06 06 06 | 00 00 01 02 06 06 06 06 | 05 | 0.208 |
| 1 035 | 9 | 03 05 07 07 08 09 09 09 | 00 00 00 00 01 02 07 09 | 00 00 00 00 05 09 09 09 | 00 00 01 03 07 09 09 09 | 12 | 0.333 |
| 1 036 | 9 | 00 00 00 00 06 09 09 09 | 00 01 02 02 03 07 09 09 | 00 01 04 09 09 09 09 09 | 00 00 05 08 09 09 09 09 | 24 | 0.667 |
| 1 037 | 9 | 00 02 04 06 09 09 09 09 | 00 00 00 00 00 01 05 09 | 00 00 00 01 07 09 09 09 | 00 00 01 02 06 09 09 09 | 08 | 0.222 |
| 1 038 | 9 | 00 01 02 04 08 08 09 09 | 00 00 00 00 00 01 04 09 | 00 01 01 01 03 09 09 09 | 00 01 01 02 06 08 09 09 | 06 | 0.167 |
| 1 039 | 9 | 00 02 02 02 05 08 09 09 | 00 00 00 02 06 08 09 09 | 00 00 00 01 04 09 09 09 | 00 02 02 06 08 09 09 09 | 17 | 0.472 |
| 1 040 | 9 | 00 00 01 03 05 07 09 09 | 00 00 00 00 01 07 09 09 | 00 00 00 01 04 09 09 09 | 00 00 00 01 07 09 09 09 | 10 | 0.278 |
| 0 041 | 9 | 00 01 05 07 09 09 09 09 | 00 00 00 00 00 01 02 09 | 00 00 00 02 09 09 09 09 | 00 00 00 02 06 09 09 09 | 10 | 0.278 |
| 1 042 | 9 | 00 00 00 01 07 09 09 09 | 00 01 01 02 02 04 05 09 | 00 00 01 01 03 08 09 09 | 01 01 02 03 09 09 09 09 | 08 | 0.222 |
| 1 043 | 9 | 00 01 03 06 09 09 09 09 | 00 00 00 00 01 02 08 09 | 00 00 01 05 09 09 09 09 | 00 00 00 00 06 08 09 09 | 10 | 0.278 |
| 0 044 | 3 | 00 00 00 01 02 02 02 03 | 00 00 00 00 00 00 01 03 | 00 01 01 01 02 03 03 03 | 00 00 00 00 01 02 03 03 | 01 | 0.083 |
| 1 045 | 9 | 00 00 00 03 06 08 09 09 | 00 00 00 00 02 04 08 09 | 00 00 00 03 07 09 09 09 | 00 00 02 02 08 09 09 09 | 09 | 0.250 |
| 0 046 | 9 | 00 00 03 03 07 08 09 09 | 00 00 00 00 00 00 00 07 | 00 00 00 00 01 07 09 09 | 00 00 00 00 04 06 09 09 | 03 | 0.083 |
| 1 047 | 9 | 00 01 02 03 07 09 09 09 | 00 00 01 01 03 06 06 09 | 00 00 00 01 07 09 09 09 | 00 00 01 01 07 09 09 09 | 10 | 0.278 |
| 1 048 | 6 | 00 00 00 01 05 05 05 06 | 00 00 00 00 01 02 06 06 | 00 00 00 02 05 06 06 06 | 00 00 00 00 03 06 06 06 | 04 | 0.167 |
| 1 049 | 9 | 00 00 00 02 04 06 08 09 | 00 00 00 00 00 04 08 09 | 00 00 00 02 05 07 09 09 | 00 00 00 01 04 09 09 09 | 07 | 0.194 |
| 0 050 | 9 | 00 01 03 04 08 08 08 09 | 00 00 00 00 01 01 01 09 | 00 00 00 00 00 06 09 09 | 00 00 00 00 00 01 07 09 | 04 | 0.111 |
| 1 051 | 6 | 00 02 03 05 05 05 06 06 | 00 00 00 00 00 00 05 06 | 00 00 00 02 04 06 06 06 | 00 01 03 04 05 06 06 06 | 09 | 0.375 |
| 1 052 | 9 | 00 00 00 01 04 07 09 09 | 00 00 00 01 01 04 06 09 | 00 00 00 00 07 09 09 09 | 00 00 00 00 05 08 09 09 | 04 | 0.111 |
| 0 053 | 7 | 00 00 00 04 06 07 07 07 | 00 01 01 01 01 01 07 07 | 00 00 00 00 03 07 07 07 | 00 00 00 00 04 07 07 07 | 01 | 0.036 |
| 1 054 | 9 | 00 00 00 01 04 07 09 09 | 00 00 00 01 01 02 05 09 | 00 01 02 05 08 09 09 09 | 00 00 00 03 07 09 09 09 | 10 | 0.278 |
| 1 055 | 9 | 00 01 01 03 04 06 09 09 | 00 00 00 00 00 00 05 08 | 00 01 01 05 08 09 09 09 | 00 00 00 01 05 08 08 09 | 07 | 0.194 |
| 1 056 | 6 | 00 00 01 02 03 04 04 06 | 00 01 02 04 04 04 05 06 | 00 02 03 04 06 06 06 06 | 00 03 05 06 06 06 06 06 | 15 | 0.625 |

Class Separability Value

USE THRESH = 0.250

56 SUBJECTS

Table II-2 (Continued). Processing results summary data used in class separability analysis.

## II.9 Bibliography


Barnsley, M., *Fractals Everywhere*, Academic Press, London, 1988.

Dawkins, R., *The Blind Watchmaker*, W.W. Norton and Co., New York, 1987.

Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Co., New York, 1989.

Langton, C.G., *Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*, Vol. VI of the Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley Publishing Co., New York, 1989.

Langton, C.G., et al, *Artificial Life II: Proceedings of the Workshop on Artificial Life, Vol.* X of the Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley Publishing Co., New York, 1992.

Mandelbrot, B., *The Fractal Geometry of Nature*, W.H. Freeman and Co., New York, 1983.

Manneville, P, Boccara, N., Vichniac, G.Y., and Bidaux, R., *Cellular Automata and Modeling of Complex Physical Systems*, Springer-Verlag, New York, 1989.

Peitgen, H.O., Jurgens, H., and Saupe, D., *Chaos and Fractals: New Frontiers of Science*, Springer-Verlag, New York, 1992.

Pickover, C.A., *Computers, Pattern, Chaos and Beauty*, St. Martin's Press, New York, 1990.

Pickover, C.A., *Computers and the Imagination: Visual Adventures Beyond the Edge*, St. Martin's Press, New York, 1991.

Schroeder, M., *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*, W. H. Freeman and Co., New York, 1991.

Thompson, J.M.T. and Stewart, H.B., *Nonlinear Dynamics and Chaos*, John Wiley and Sons, New York, 1987.

Vicsek, T., *Fractal Growth Phenomena*, World Scientific, Singapore, 1989.

# III. Summary and Conclusions

Data processing was a major component of this effort, leading to the compilation of a database of 56 confirmed subjects, 41 of which were confirmed deceptive, and 15 were confirmed as non deceptive. These subjects were extracted from a database of 484 possible subjects, 129 of which were confirmed. This low yield (56 of 129 successfully compiled into a database for analysis) was a result of several factors: inability to read the Axciton files; corrupted or incomplete event marker files; inability to correlate question file with event markers; unknown compressed format for certain subjects; and missing subjects (in all, 113 documented subjects were not included in the 90mB disk supplied by APL). General conclusions cannot be reached from such a relatively small database, but it was sufficient for studying the structure of features that allow for classification of polygraphs. We have attempted to resolve the difficulties with the raw data by contacting APL, but to date, we have not been successful. Future work in artificial neural network processing of polygraph signals will require a substantially larger database for the purpose of training and validation of scoring accuracy.

A novel data representation and processing approach has been developed, involving representation of polygraph channel data (CR - pairs) as phase plots which are in turn analyzed using cellular automata (CA). This approach is mainly aimed at extracting relevant features from the channels that can be used for accurate classification of the polygraph. In an on going parallel study (N00014-93-C-0207, Design and Training of an Artificial Neural Network for Polygraph Signal Processing), the features extracted via the CAs will be analyzed and the polygraph scored using an artificial neural network. However, an analysis of the class separability of the features extracted by the CA alone has yielded promising results: based on the current database, the CA can correctly classify 95% of the deceptive subjects, and correctly classify 87% of the non deceptive subjects. While these results are encouraging and clearly show the potential usefulness of neural network methods in polygraphy, further development of post feature extraction classifiers (e.g. the artificial neural network) and extensive evaluation against a much larger database of confirmed subjects is required in order to demonstrate and validate a classifier that could be trusted and certified for general use by polygraph examiners.