AD-A244 149

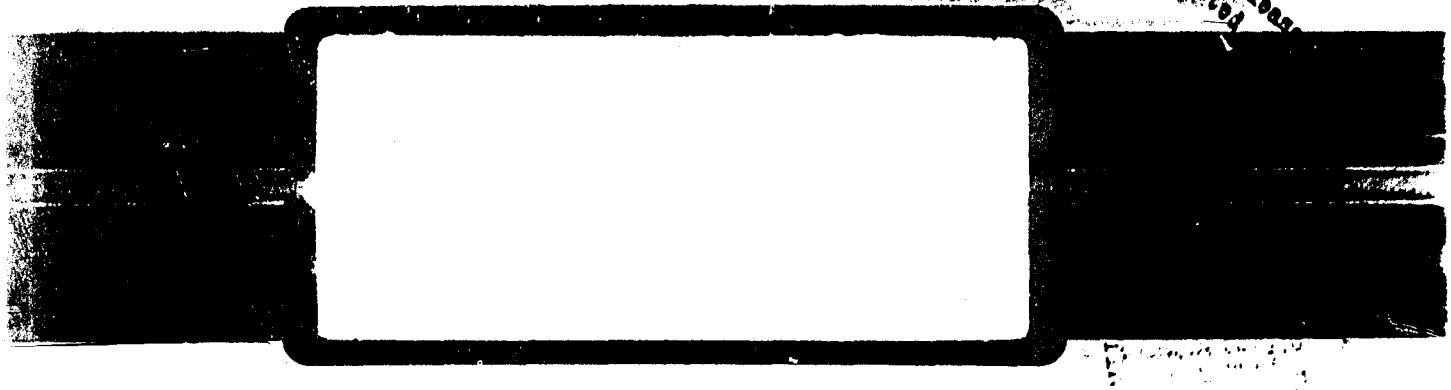


DTIC
ELECTE
DEC 26 1991
S C D

(1)

AFOSR-TR- 91 0962

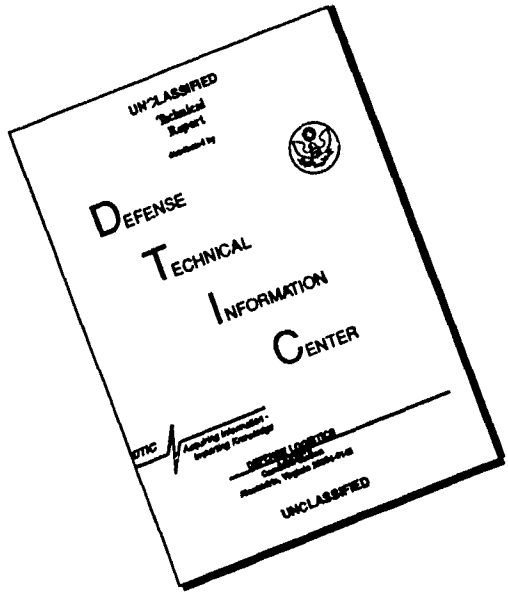
Approved for public release;
distribution unlimited.



Universal Energy Systems, Inc.



DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	18 April 1991	Final 1 Feb 88-29 Sep 90

4. TITLE AND SUBTITLE
United States Air Force
High School Apprenticeship Program
1990 Program Management Report ~~(w/atch 1,2, & 3)~~ Volume I of IV

5. FUNDING NUMBERS
F49620-88-C-0053

6. AUTHOR(S)
Mr Rodney Darrah

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Universal Energy Systems, Inc. (UES)
4401 Dayton-Xenia Road
Dayton OH 45432-1894

8. PERFORMING ORGANIZATION REPORT NUMBER
AFOSR-TR- 91 0902

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NI
Bldg 410
Bolling AFB DC 20332-6448
Lt Col V. Claude Cavender

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Air Force High School Apprenticeship Program's purpose is to place outstanding high school students whose interests are in the areas of mathematics, engineering, and science to work in a laboratory environment. The students selected to participate work in an Air Force Laboratory for a duration of 8 weeks during their summer vacation.

91-18946

14. SUBJECT TERMS
91 1223 202

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLAAIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

UNITED STATES AIR FORCE

HIGH SCHOOL APPRENTICESHIP PROGRAM

1990

PROGRAM MANAGEMENT REPORT

VOLUME I OF IV

UNIVERSAL ENERGY SYSTEMS, INC.

Program Director, UES
Rodney C. Darrah

Program Administrator, UES
Susan K. Espy

Program Manager, AFOSR
Lt. Col. Claude Cavender

Submitted to

Air Force Office of Scientific Research

Bolling Air Force Base

Washington, DC

December 1990

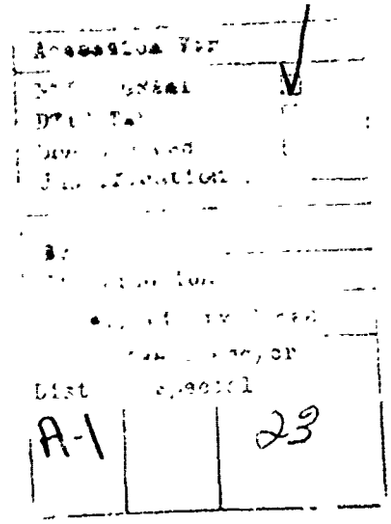


TABLE OF CONTENTS

SECTION	PAGE
INTRODUCTION	1
TABLE 1 - CALENDAR OF ACTIVITIES	2
RECRUITING AND SELECTION	3
PARTICIPANT LABORATORY ASSIGNMENT	4
INFORMATION PACKAGE	7
MODEL PLACEMENT LETTER TO STUDENT	8
STUDENT ACCEPTANCE FORM	9
MENTOR EVALUATION QUESTIONNAIRE AND RESPONSES	10
APPRENTICE EVALUATION QUESTIONNAIRE AND RESPONSES	57
LIST OF PARTICIPANTS FINAL REPORTS	64
FINAL REPORTS	72

INTRODUCTION

In the near future the United States may face shortages of scientists and engineers in fields such as physics, electronic engineering, computer science and aeronautical engineering. High school students are currently not selecting to prepare for careers in these areas in numbers large enough to match the projected needs in the United States.

The Air Force faces "a formidable challenge - the acquisition and retention of the technological competence needed to ensure a strong national security, both in-house and in the industrial and academic base which supports defense preparedness." The Director of the Office and Science of Technology Policy in the Executive Office of the President in 1979 responded to this need by requesting the federal agencies to incorporate in their contract research programs the mechanisms to stimulate career interests in science and technology in high school students showing promise in these areas. The Air Force High School Apprenticeship Program is an example of the response to this.

Under the Special Studies section of the Summer Faculty Research Program an Air Force High School Apprenticeship was initiated. This program's purpose is to place outstanding high school students whose interests are in the areas of engineering and science to work in a laboratory environment. The students who were selected to participate worked in one of the Air Force Laboratories for a duration of 8 weeks during their summer vacation.

There has been a few incidents concerning misuse of the computers in the laboratories. On two separate occasions the laboratory has had to revoke computer privileges on four high school students. Both of these incidents happened at the same laboratory.

Two years ago two students wrote a program to shut down the computer system at the laboratory and to steal users access codes. These students were removed from the laboratory and one of the students was dismissed from the program, while the other student finished his apprenticeship at the UES facility.

This year a similar incident happened. Two other students were involved with basically the same incident. One student wrote a program that would send repeating messages to the various computer terminals. Included in this program was a password interception program were the users would type in their password and the program would retrieve that password. The other student involved also wrote and executed a password stealing program, and was involved in the unauthorized use of a government computer in writing a fraudulent letter. The student also obtained unauthorized access to a computer modem.

The Air Force High School Apprenticeship Program was modeled after the Army's High School Program, which is very successful.

The following time schedule was used in order to accomplish this effort.

TABLE 1
AIR FORCE HIGH SCHOOL
APPRENTICESHIP PROGRAM

Calendar of Activities

December	o Identify schools and laboratories for participation o Prepare informational material for schools and installations application forms for students and mentors, and covering letters. o Disseminate information o Recruit apprentices, mentors
January	o Send student applications to teachers
February	o Applications with teacher recommendations o Receive mentors' project descriptions and student requirements o Make preliminary selection of students for referral to mentor
March	o Make preliminary matching of students with mentors; send letters with several student applications to each mentor o Mentors interview students, inform UES of choice
April	o Send letters of placement to students, with acceptance forms to be signed by them and parents and returned to UES o Place 2nd year apprentices o Make final matches o See that security clearances are started, where applicable o (Mentors provide background reference material to chosen apprentices) o Encourage enrichment activities: arrange for films, speakers, tours, etc.
May	o Send letters to students and mentors re-opening session o Send students Apprentice Handbook
June	o Arrange general orientation for students and mentors
July, August	o Administer and monitor apprenticeships o Check on enrichment activities o Distribute evaluation forms to students and mentors
September	o Analyze evaluations o Prepare final report to Air Force

RECRUITING AND SELECTION

Application packages and the flyer were distributed to the laboratories and to the various high schools within convenient driving distance of the laboratories (typically less than 20 miles).

There was a total of 516 applications received by UES on the program. When the applications were received, a copy was sent to the appropriate laboratory for review. The laboratory mentor screened the applications and conducted personnel interviews with the high school students then sent UES a prioritized list of their applicants. There were a total of 132 participants on the program, selected from the 516 applications.

The laboratories participating in the program along with the number of students assigned to the laboratory is listed below:

Aero Propulsion Laboratory	7
Armament Laboratory	16
Armstrong Aerospace Medical Research Laboratory	7
Arnold Engineering and Development Center	6
Avionics Laboratory	6
Astronautics Laboratory	12
Engineering and Services Center	15
Electronic Technology Laboratory	5
Flight Dynamics Laboratory	9
Geophysics Laboratory	7
Materials Laboratory	1
Occupational and Environmental Health Laboratory	3
Rome Air Development Center	15
School of Aerospace Medicine	13
Weapons Laboratory	10

Participant Laboratory Assignment
1990 High School Apprenticeship Program

Aero Propulsion Laboratory
Wright-Patterson Air Force Base, Ohio

1. Matthew Bold
2. Hee Sun Choung
3. Katharine Day
4. Chris Hatch
5. Chet Nieter
6. Jennifer Pollock
7. Carol Rogers

Armament Laboratory
Eglin Air Force Base, Florida

1. Steven Bryan
2. Toyna Cook
3. Heather Cox
4. Kathryn Deibler
5. Chris Ellis
6. Dana Farver
7. Kenneth Gage
8. Reid Harrison
9. Derek Holland
10. Christine Riendeau
11. Lisa Schmidt
12. Patricia Tu
13. Troy Urquhart
14. Gregory VanWiggeren
15. Danielle Walker
16. Eric White

Armstrong Aerospace Medical Research Laboratory
Wright-Patterson Air Force Base, Ohio

1. Rex Ballinger
2. Douglas Brungart
3. Caroline Chuang
4. Keisha Hayes
5. Douglas Marshak
6. Jeremiah Rogers
7. James Shamiyeh

Arnold Engineering and Development Center
Arnold Air Force Base, Tennessee

1. Timothy Craddock
2. Myra Medley
3. Julie Reece
4. Jonathan Sanders
5. Jason Scott
6. Gerald Turner

Astronautics Laboratory
Edwards Air Force Base, California

1. Alisha Conrow
2. Debra Meyer
3. John Moro
4. Lloyd Neurauter
5. Joseph Padilla
6. Melanie Pyle
7. Thomas Quinn
8. Tracy Reed
9. Benjamin Sommers
10. Stephanie VanMeter
11. Rebecca Weston
12. David Youmans

Avionics Laboratory
Wright-Patterson Air Force Base, Ohio

1. Brian Barclay
2. Mark Boeke
3. Michael Chabinyc
4. David Collins
5. Austin Flack
6. Jerard Wilson

Engineering and Services Center
Tyndall Air Force Base, Florida

1. Jennifer Brewer
2. Philip Dorsch
3. David Eshleman
4. Richard Hartzer
5. Thor Johnson
6. Tracy Lamb
7. Brent Miller
8. Debra Piechowiak
9. Jonathan Protz
10. Julie Scruggs
11. Michael Stone
12. Amy Thomas
13. Michael Thompson
14. Jeffrey Ward
15. Robin Woodworth

Electronic Technology Laboratory
Wright-Patterson Air Force Base, Ohio

1. Matthew Brewer
2. Matt Elwood
3. Shelly Knupp
4. Christopher O'Dell
5. Suzette Yu

Flight Dynamics Laboratory
Wright-Patterson Air Force Base, Ohio

1. Jean Ay
2. Matthew Becker
3. Wendy Choate
4. Andrea Dean
5. Rachael Lyon
6. Cathie Moore
7. Roderick Morgan
8. Stanley Wall
9. Douglas Wickert

Geophysics Laboratory
Hanscom Air Force Base, Massachusetts

1. Stephen Britten
2. Weihaw Chuang
3. Christopher Guild
4. Jason Klingensmith
5. Galen McKinley
6. Jeffrey Sayasane
7. Paul Swietek

Materials Laboratory
Wright-Patterson Air Force Base, Ohio

1. Jennifer Walker

Occupational and Environment Health Laboratory
Brooks Air Force Base, Texas

1. Gary New
2. Andrea Perez
3. Michael Smid

Rome Air Development Center
Griffiss Air Force Base, New York

1. Daniel Abbis
2. Mark Anania
3. Bridget Bordiuk
4. Todd Gleason
5. Christopher Hailes
6. Edward Holmes
7. Kimberly King
8. Kathryn Lee
9. Jason Lenio
10. Kevin Olson
11. David Petrillo
12. Thomas Potter
13. Daniel Russell
14. Philip Schremmer
15. Eric Shaw

School of Aerospace Medicine
Brooks Air Force Base, Texas

1. Anthony Barnes
2. Whitney Brandt
3. Deann Cooper
4. Matthew Felder
5. Christopher Hudson
6. Sonya Longbotham
7. Brian McBurnett
8. Heather Neville
9. Lori Olenick
10. Joanna Saucedo
11. Wendy Shields
12. Brent Strawn
13. John Taboada

Weapons Laboratory
Kirtland Air Force Base, New Mexico

1. David Cochrell
2. Gregory Hays
3. David Knapp
4. Aaron Laing
5. Kerim Martinez
6. Ryan McAlhaney
7. Margaret Morecock
8. Philip Ortiz
9. Brian Rizzoli
10. Chris Stoltenberg

INFORMATION PACKAGE

23 March 1990

Dear :

Enclosed are the mentor applications forms for the 1990 USAF High School Apprenticeship Program. The mentors and project descriptions have been approved by UES.

Enclosed are the applications for the High School Apprenticeship program for the summer of 1990. The following mentors and previous high school participants have been matched and selected to work with each other for the coming summer.

	Student	Mentor
1.		
2.		
3.		

The following is a previous high school participant in the program and is selected to participate in the program for this summer. He needs to be matched with one of the approved mentors for this summer.

	Student
1.	

The remainder of the students need to be evaluated by the approved mentors for possible selection in the program for this summer. Please provide to UES a listing of the mentor recommendations for students by 15 April 1990.

We have a total of 100 positions available on the program for this summer. We will select as many as possible to fill this available positions. We anticipate that about 15 high school students will be selected to participate with the mentors at the Rome Air Development Center.

If you have any questions concerning this information, please do not hesitate to contact us.

Sincerely,

UNIVERSAL ENERGY SYSTEMS, INC.

Rodney C. Darrah
Program Director

Enclosure

xc: Lt. Col. Claude Cavender

MODEL PLACEMENT LETTER TO STUDENT

13 March 1991

1~
2~
3~

Dear 4~:

Congratulations! You have been selected to participate in the Air Force Office of Scientific Research High School Apprenticeship Program as an apprentice to 5~ at the 6~ to work on Project: "7~" from June 18 to August 10, 1990. Enclosed is an acceptance form for you and your parent or guardian to sign. Also enclosed is your W-4 form which needs to be filled out and returned along with your acceptance form to me by May 11, 1990.

The Apprenticeship Program provides an exciting opportunity for you, and we hope you will take advantage of the work experience to learn more about scientific research, career opportunities in science and engineering, and the education necessary to prepare yourself for such careers. On June 18, 1990, the first day of the program, you are expected to attend an orientation session with other apprentices and mentors and to ask questions about any concerns you might have. Many of those concerns are discussed in the Apprentice Handbook which is enclosed. The Handbook also contains suggestions for getting the most out of the summer experience, and references to other work experience programs and financial assistance available for college education. Please read the Handbook before the orientation session, so that time will not be used for questions answered in the book.

You will be expected to begin work promptly at 8:00 a.m. on June 18. If for any reason you cannot begin work on that day, or cannot report to work on any future work day, you must inform your mentor at 8~.

We hope you will enjoy your apprenticeship. I will be available throughout the summer should problems arise that cannot be solved by your mentor.

Sincerely,

UNIVERSAL ENERGY SYSTEMS, INC.

Rodney C. Darrah
Program Director

RCD/mt

STUDENT ACCEPTANCE FORM

for participation in

Air Force Office of Scientific Research

High School Apprenticeship Program, 1990

I, 1~, accept the position of apprentice in the Air Force Office of Scientific Research High School Apprenticeship Program from June 18, 1990 to August 10, 1990 to work with 2~ at the 3~ on Project: "4~". I understand that I will receive a stipend of $5~ for the summer apprenticeship for which I must participate during the entire session.

Date _____ Signature of student _____

High School _____

PARENT CONSENT

As the parent/guardian, I certify that my son/daughter/ward has my permission to participate in this project for secondary school students. It is my understanding that he/she will be subject to the regulations of the host institution and the project. I understand that should a health emergency arise I will be notified, but that if I cannot be reached by telephone, such medical treatment as deemed necessary by competent medical personnel is authorized.

Date _____ Signature of parent _____

Daytime phone _____

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

1. How did you hear about program?

   o Previous mentor
   o Notice on bulletin board
   o Memo from personnel office
   o Verbal request from personnel office
   o Other, specify_____
   _____

2. Did you volunteer to be a mentor?

   Yes___ No___

3. Did the student application provide sufficient information?

   Yes___ No___

4. If no, what additional information would you want to see included on the student application form?_____

   _____

   _____

5. Did you interview the student who was placed in your laboratory before the program started?

   Yes___ No___

6. If no, would an interview have been useful?

   Yes___ No___

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   A lot___ Some___ Not at all___

8. How much did the student contribute to the research of your laboratory?

   A lot___ Some___ Not at all___

9. How would you rate the student's performance?

   Excellent___ Fair___ Poor___

10. Would like to participant as a mentor for the program next summer?

Yes__ No__ If No, Why?_____

11. Would you want the same student in your laboratory next summer?

Yes__ No__ If No, Why?_____

12. Did the work of the student influence his/her choice of

a. courses in coming school year? __Yes __No __Don't know

Explain _____

b. career choice? __Yes __No __Don't know

Explain _____

If you have suggestions or comments on the program, please use the space below.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

_____
Name of student apprentice

_____
Name of mentor/laboratory

_____
Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

AERO PROPULSION LABORATORY

1. How did you hear about program?

   6 Previous mentor
   0 Notice on bulletin board
   0 Memo from personnel office
   0 Verbal request from personnel office
   1 Other, specify:_____

   Assistant Chief Scientists.

2. Did you volunteer to be a mentor?

   7 Yes
   0 No

3. Did the student application provide sufficient information?

   7 Yes
   0 No
   0 Don't Know

4. If no, what additional information would you want to see included on the student application form?

   Specific information regarding computer experience.

5. Did you interview the student who was placed in your laboratory before the program started?

   4 Yes
   3 No

6. If no, would an interview have been useful?

   6 Yes
   1 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   6 A lot
   1 Some
   0 Not at all

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   6 A lot
   1 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   2 A lot
   5 Some
   0 Not at all

9. How would you rate the student's performance?

   4 Excellent
   3 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    7 Yes
    0 No
      If No, Why?

11. Would you want the same student in your laboratory next summer?

    7 Yes
    0 No
      If No, Why?

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 0 - Yes 1 - No 6 - Don't Know

       Explain:

       She was already planning on an Engineering degree at U. of K.

    b. career choice? 0 - Yes 1 - No 6 - Don't know

       Explain:

If you have suggestions or comments on the program, please use the space below.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

Name of student apprentice

Name of mentor/laboratory

Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

ARMAMENT LABORATORY

1. How did you hear about program?

   11 Previous mentor
   0 Notice on bulletin board
   3 Memo from personnel office
   0 Verbal request from personnel office
   2 Other, specify:_____

   Section Chief.

   I have been a mentor for 3 years.

2. Did you volunteer to be a mentor?

   16 Yes
   0 No

3. Did the student application provide sufficient information?

   12 Yes
   1 No
   3 N/A

4. If no, what additional information would you want to see included on the student application form?

   I did not see the application, or pay much attention to it. I just accepted the student assigned to me.

5. Did you interview the student who was placed in your laboratory before the program started?

   10 Yes
   6 No

6. If no, would an interview have been useful?

   5 Yes
   0 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   15 A lot
   1 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   10 A lot
   6 Some
   0 Not at all

9. How would you rate the student's performance?

   15 Excellent
   1 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    14 Yes
    2 No
    If No, Why?

    I will be away.

11. Would you want the same student in your laboratory next summer?

    10 Yes
    6 No
    If No, Why?

    All the no responses indicate the students will be attending college and not eligible for the program.

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 5 - Yes 2 - No 9 - Don't know

    Explain:

    Most of the comments indicate that the courses are already set. But toward the math and science courses.

b. career choice? 7 - Yes 1 - No 8 - Don't know

Explain:

The comments consist that students are still deciding, two of the students definitely want in the science careers.

If you have suggestions or comments on the program, please use the space below.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

Name of student apprentice

Name of mentor/laboratory

Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY

1. How did you hear about program?

   7 Previous mentor
   0 Notice on bulletin board
   0 Memo from personnel office
   0 Verbal request from personnel office
   0 Other, specify:_____

2. Did you volunteer to be a mentor?

   7 Yes
   0 No

3. Did the student application provide sufficient information?

   6 Yes
   0 No

4. If no, what additional information would you want to see included on the student application form?

   I didn't see the application form.

5. Did you interview the student who was placed in your laboratory before the program started?

   2 Yes
   5 No

6. If no, would an interview have been useful?

   2 Yes
   2 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   3 A lot
   4 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   2 A lot
   5 Some
   0 Not at all

9. How would you rate the student's performance?

   6 Excellent
   1 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    4 Yes
    0 No
      If No, Why?

11. Would you want the same student in your laboratory next summer?

    6 Yes
    1 No
      If No, Why?

    She graduated.

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 2 - Yes 1 - No 4 - Don't know

       Explain:

       Gained additional knowledge & training in lab that allowed testing out of some pre-requisite courses.

    b. career choice? 2 - Yes 2 - No 3 - Don't know

       Explain:

       Comments include that the student has new insight to engineering, and another student wants to go into medicine

If you have suggestions or comments on the program, please use the space below.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

Name of student apprentice

Name of mentor/laboratory

Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

ARNOLD ENGINEERING AND DEVELOPMENT CENTER

1. How did you hear about program?

   1 Previous mentor
   0 Notice on bulletin board
   2 Memo from personnel office
   2 Verbal request from personnel office
   1 Other, specify:_____

   Supervisor.

2. Did you volunteer to be a mentor?

   5 Yes
   1 No

3. Did the student application provide sufficient information?

   6 Yes
   0 No

4. If no, what additional information would you want to see included on the student application form?

5. Did you interview the student who was placed in your laboratory before the program started?

   4 Yes
   2 No

6. If no, would an interview have been useful?

   1 Yes
   1 No
   0 Maybe

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   4 A lot
   2 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   4 A lot
   2 Some
   0 Not at all

9. How would you rate the student's performance?

   6 Excellent
   0 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    6 Yes
    0 Maybe
    0 No
      If No, Why?

11. Would you want the same student in your laboratory next summer?

    6 Yes
    0 No
      If No, Why?

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 1 - Yes 3 - No 2 - Don't know

       Explain:

       The no responses indicate that courses are pre-determined and not many options.

    b. career choice? 4 - Yes 0 - No 2 - Don't know

       Explain:

       Heightened interest in chemistry/chemical engineering.

       Solidified his intent to pursue an engineering career.

If you have suggestions or comments on the program, please use the space below.

I would suggest that students requiring a security clearance he given advanced notice so that the necessary processing could be completed prior to their coming to work. A ten week program (instead of 8) should be offered as an option for the students.

This program was a very positive experience for me as well as her. I would enjoy participating in the program again.

PLEASE RETURN BY 14 September 1990

_____
Name of student apprentice

to: Susan Espy
Coordinator

_____
Name of mentor/laboratory

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

_____
Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

ASTRONAUTICS LABORATORY

1. How did you hear about program?

   5 Previous mentor
   0 Notice on bulletin board
   1 Memo from personnel office
   1 Verbal request from personnel office
   2 Other, specify:_____

   Request from XRX.

2. Did you volunteer to be a mentor?

   9 Yes
   0 No

3. Did the student application provide sufficient information?

   8 Yes
   1 No

4. If no, what additional information would you want to see included on the student application form?

   At a high school level there isn't a lot of detailed scientific technical questions you can ask for.

5. Did you interview the student who was placed in your laboratory before the program started?

   4 Yes
   5 No

6. If no, would an interview have been useful?

   4 Yes
   1 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   7 A lot
   2 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   6 A lot
   3 Some
   0 Not at all

9. How would you rate the student's performance?

   9 Excellent
   0 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    9 Yes
    0 No
      If No, Why?

11. Would you want the same student in your laboratory next summer?

    9 Yes
    0 No
      If No, Why?

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 2 - Yes 4 - No 3 - Don't know

       Explain:

       Most of the responses indicate that student's courses are already set for the upcoming year.

    b. career choice? 1 - Yes 3 - No 5 - Don't know

       Explain:

       Two of the comments were that the student's have their career's planned, even as far as job opportunities. One student wants to go in the medical profession.

If you have suggestions or comments on the program, please use the space below.

Let the students accrue leave (annual & sick) and let them work more than 40 days!

PLEASE RETURN BY 14 September 1990

_____
Name of student apprentice

to: Susan Espy
Coordinator

_____
Name of mentor/laboratory

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

_____
Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

AVIONICS LABORATORY

1. How did you hear about program?

   3 Previous mentor
   1 Notice on bulletin board
   1 Memo from personnel office
   1 Verbal request from personnel office
   0 Other, specify:_____

2. Did you volunteer to be a mentor?

   6 Yes
   0 No

3. Did the student application provide sufficient information?

   4 Yes
   2 No

4. If no, what additional information would you want to see included on the student application form?

   Previous police record and descriptions of any court imposed fines or punishment.

5. Did you interview the student who was placed in your laboratory before the program started?

   0 Yes
   6 No

6. If no, would an interview have been useful?

   2 Yes
   4 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   2 A lot
   4 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   4 A lot
   2 Some
   0 Not at all

9. How would you rate the student's performance?

   5 Excellent
   0 Fair
   1 Poor

10. Would like to participant as a mentor for the program next summer?

    5 Yes
    1 No
      If No, Why?

11. Would you want the same student in your laboratory next summer?

    5 Yes
    1 No
      If No, Why?

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 1 - Yes 3 - No 2 - Don't know

       Explain:

       I think his courses were pretty well planned out before he came to the lab. What he learned here probably reinforced his choices rather than changing them.

    b. career choice? 2 - Yes 2 - No 2 - Don't know

       Explain:

       All of the responses indicate the students' have chosen a career; from chemical engineer to computer science.

If you have suggestions or comments on the program, please use the space below.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

Name of student apprentice

Name of mentor/laboratory

Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

ENGINEERING AND SERVICES CENTER

1. How did you hear about program?

   9 Previous mentor
   1 Notice on bulletin board
   0 Memo from personnel office
   1 Verbal request from personnel office
   4 Other, specify:_____

   The comments for the Other category were notifications from AFESC staff.

2. Did you volunteer to be a mentor?

   14 Yes
   1 No

3. Did the student application provide sufficient information?

   14 Yes
   1 No

4. If no, what additional information would you want to see included on the student application form?

   Never saw a student application from.

5. Did you interview the student who was placed in your laboratory before the program started?

   4 Yes
   11 No

6. If no, would an interview have been useful?

   8 Yes
   4 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   9 A lot
   5 Some
   1 Not at all

8. How much did the student contribute to the research of your laboratory?

   9 A lot
   5 Some
   1 Not at all

9. How would you rate the student's performance?

   14 Excellent
   1 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    13 Yes
    2 No
      If No, Why?

    The "no" comments were because of the time that it takes, and the other mentor will be traveling.

11. Would you want the same student in your laboratory next summer?

    15 Yes
    0 No
      If No, Why?

    If I were to do it again.

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 5 - Yes 4 - No 6 - Don't know

       Explain:

       The majority of the comments as before where that the courses are already determined.

    b. career choice? 4 - Yes 3 - No 8 - Don't know

       Explain:

       The comments range from students that have not decided, to a student choosing to be an engineer and Air Force pilot.

If you have suggestions or comments on the program, please use the space below.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

Name of student apprentice

Name of mentor/laboratory

Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

ELECTRONIC TECHNOLOGY LABORATORY

1. How did you hear about program?

   0 Previous mentor
   0 Notice on bulletin board
   1 Memo from personnel office
   0 Verbal request from personnel office
   4 Other, specify:_____

   Through Lab Operation Division (ELA).

   Verbal request from boss.

   Co-worker.

2. Did you volunteer to be a mentor?

   4 Yes
   1 No

3. Did the student application provide sufficient information?

   5 Yes
   0 No

4. If no, what additional information would you want to see included on the student application form?

5. Did you interview the student who was placed in your laboratory before the program started?

   3 Yes
   2 No

6. If no, would an interview have been useful?

   1 Yes
   1 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   5 A lot
   0 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   3 A lot
   2 Some
   0 Not at all

9. How would you rate the student's performance?

   4 Excellent
   1 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    5 Yes
    0 No
      If No, Why?

11. Would you want the same student in your laboratory next summer?

    4 Yes
    1 No
      If No, Why?

    Student has limited interest in research.

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 0 - Yes 1 - No 4 - Don't know

       Explain:

       Responses indicate that courses are pre-determined.

    b. career choice? 0 - Yes 0 - No 5 - Don't know

       Explain:

       I think she has gained an appreciation for the challenging nature of research.

If you have suggestions or comments on the program, please use the space below.

PLEASE RETURN BY 14 September ' 90

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

Name of student apprentice

Name of mentor/laboratory

Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

FLIGHT DYNAMICS LABORATORY

1. How did you hear about program?

   4 Previous mentor
   0 Notice on bulletin board
   0 Memo from personnel office
   1 Verbal request from personnel office
   3 Other, specify:_____

   Branch office.

   WRDC/FIOP

2. Did you volunteer to be a mentor?

   8 Yes
   0 No

3. Did the student application provide sufficient information?

   8 Yes
   0 No

4. If no, what additional information would you want to see included on the student application form?

5. Did you interview the student who was placed in your laboratory before the program started?

   3 Yes
   5 No

6. If no, would an interview have been useful?

   4 Yes
   1 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   5 A lot
   3 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   5 A lot
   3 Some
   0 Not at all

9. How would you rate the student's performance?

   7 Excellent
   1 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    7 Yes
    1 No
      If No, Why?

    Only if a project is available for use by student!

11. Would you want the same student in your laboratory next summer?

    6 Yes
    2 No
      If No, Why?

    Comments indicate student's are not eligible.

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 1 - Yes 2 - No 5 - Don't know

       Explain:

       Student had already chosen Engineering Curriculum for College.

    b. career choice? 3 - Yes 1 - No 4 - Don't know

       Explain:

       Student already targets Aerospace future.

If you have suggestions or comments on the program, please use the space below.

Work must be available that a student can get involved in for the duration of there stay. If it is not, I will not take any more students.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

_____
Name of student apprentice

_____
Name of mentor/laboratory

_____
Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

GEOPHYSICS LABORATORY

1. How did you hear about program?

   2 Previous mentor
   0 Notice on bulletin board
   3 Memo from personnel office
   0 Verbal request from personnel office
   0 Other, specify:_____

2. Did you volunteer to be a mentor?

   5 Yes
   0 No

3. Did the student application provide sufficient information?

   5 Yes
   0 No

4. If no, what additional information would you want to see included on the student application form?

5. Did you interview the student who was placed in your laboratory before the program started?

   1 Yes
   4 No

6. If no, would an interview have been useful?

   3 Yes
   1 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   3 A lot
   2 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   3 A lot
   2 Some
   0 Not at all

9. How would you rate the student's performance?

   5 Excellent
   0 Fair
   0 Poor

10. Would like to participate as a mentor for the program next summer?

    5 Yes
    0 No
      If No, Why?

11. Would you want the same student in your laboratory next summer?

    5 Yes
    0 No
      If No, Why?

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 0 - Yes 0 - No 5 - Don't know

       Explain:

       Courses probably selected prior to the summer job.

    b. career choice? 0 - Yes 1 - No 4 - Don't know

       Explain:

       Was already planning to enter MIT in an engineering field.

If you have suggestions or comments on the program, please use the space below.

The main comment is that they would like to see the program expanded to 10 to 12 weeks, also that the stipend should be raised to compete with jobs outside of research.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

Name of student apprentice

Name of mentor/laboratory

Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

MATERIALS LABORATORY

1. How did you hear about program?

   0 Previous mentor
   0 Notice on bulletin board
   0 Memo from personnel office
   1 Verbal request from personnel office
   0 Other, specify:_____

2. Did you volunteer to be a mentor?

   1 Yes
   0 No

3. Did the student application provide sufficient information?

   1 Yes
   0 No

4. If no, what additional information would you want to see included on the student application form?

5. Did you interview the student who was placed in your laboratory before the program started?

   0 Yes
   1 No

6. If no, would an interview have been useful?

   1 Yes
   0 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   1 A lot
   0 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   1 A lot
   0 Some
   0 Not at all

9. How would you rate the student's performance?

   1 Excellent
   0 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    1 Yes
    0 No
    If No, Why?

11. Would you want the same student in your laboratory next summer?

    1 Yes
    0 No
    If No, Why?

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 0 - Yes 0 - No 1 - Don't know

       Explain:

       Student is pursuing a career in Civil Engineering.

    b. career choice? 0 - Yes 1 - No 0 - Don't know

       Explain:

If you have suggestions or comments on the program, please use the space below.

She was really a pleasure to work with. She was self motivated and a diligent worker who was genuinely interested in everything going on within the lab. She did a outstanding job. Would like to see her back next year!

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

Name of student apprentice

Name of mentor/laboratory

Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

OCCUPATIONAL AND ENVIRONMENT HEALTH LABORATORY

1. How did you hear about program?

   0 Previous mentor
   0 Notice on bulletin board
   1 Memo from personnel office
   0 Verbal request from personnel office
   1 Other, specify:_____

   My supervisor.

2. Did you volunteer to be a mentor?

   1 Yes
   1 No

3. Did the student application provide sufficient information?

   2 Yes
   0 No

4. If no, what additional information would you want to see included on the student application form?

5. Did you interview the student who was placed in your laboratory before the program started?

   0 Yes
   2 No

6. If no, would an interview have been useful?

   2 Yes
   0 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   0 A lot
   2 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   2 A lot
   0 Some
   0 Not at all

9. How would you rate the student's performance?

   2 Excellent
   0 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    2 Yes
    0 No
      If No, Why?

11. Would you want the same student in your laboratory next summer?

    2 Yes
    0 No
      If No, Why?

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 0 - Yes 0 - No 2 - Don't know

       Explain:

       Student was already interested in science courses.

    b. career choice? 0 - Yes 0 - No 2 - Don't know

       Explain:

       Student is already set to take engineering in college.

If you have suggestions or comments on the program, please use the space below.

PLEASE RETURN BY 14 September 1990

_____
Name of student apprentice

to: Susan Espy
Coordinator

_____
Name of mentor/laboratory

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

_____
Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

ROME AIR DEVELOPMENT CENTER

1. How did you hear about program?

   4 Previous mentor
   0 Notice on bulletin board
   3 Memo from personnel office
   2 Verbal request from personnel office
   2 Other, specify:_____

   Director of Photonics Labs.

   Branch Chief sent me a memo.

2. Did you volunteer to be a mentor?

   10 Yes
   1 No

3. Did the student application provide sufficient information?

   10 Yes
   1 No

4. If no, what additional information would you want to see included on the student application form?

5. Did you interview the student who was placed in your laboratory before the program started?

   0 Yes
   11 No

6. If no, would an interview have been useful?

   6 Yes
   5 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   4 A lot
   7 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   3 A lot
   8 Some
   0 Not at all

9. How would you rate the student's performance?

   10 Excellent
   1 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    8 Yes
    3 No
    If No, Why?

    The "no" responses indicated that they did not have the time to devote to the students.

11. Would you want the same student in your laboratory next summer?

    8 Yes
    3 No
    If No, Why?

    Give someone else the opportunity to work here.

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 2 - Yes 2 - No 7 - Don't know

       Explain:

       Courses pre-determined but students were influenced for courses at the advanced levels.

    b. career choice? 3 - Yes 2 - No 5 - Don't know

       Explain:

       The majority of the comments were that the career choices were reinforced.

If you have suggestions or comments on the program, please use the space below.

Great Program! Thanks!

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

Name of student apprentice

Name of mentor/laboratory

Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

SCHOOL OF AEROSPACE MEDICINE

1. How did you hear about program?

   7 Previous mentor
   1 Notice on bulletin board
   0 Memo from personnel office
   1 Verbal request from personnel office
   4 Other, specify:_____

   Laboratory Chief Scientist

   Letter from SAM/CA.

2. Did you volunteer to be a mentor?

   13 Yes
   0 No

3. Did the student application provide sufficient information?

   13 Yes
   0 No

4. If no, what additional information would you want to see included on the student application form?

5. Did you interview the student who was placed in your laboratory before the program started?

   4 Yes
   8 No

6. If no, would an interview have been useful?

   4 Yes
   4 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   12 A lot
   1 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   9 A lot
   4 Some
   0 Not at all

9. How would you rate the student's performance?

   11 Excellent
   2 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    12 Yes
    0 No
      If No, Why?

11. Would you want the same student in your laboratory next summer?

    11 Yes
    2 No
      If No, Why?

    The student stated she did not have the patience for research work. Therefore, I would prefer giving another student an opportunity to participate in next year's program.

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 1 - Yes 3 - No 9 - Don't know

       Explain:

       The responses indicated that courses are pre-determined, although one comment indicated that it influenced the student s college courses.

    b. career choice? 5 - Yes 0 - No 8 - Don't know

       Explain:

       Indicated a change from nursing to biological research.

If you have suggestions or comments on the program, please use the space below.

This is an excellent program. It helps students realize what "research" means, and gives them some independence in the laboratory setting.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

_____
Name of student apprentice

_____
Name of mentor/laboratory

_____
Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
MENTOR EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A MENTOR)

WEAPONS LABORATORY

1. How did you hear about program?

   1 Previous mentor
   0 Notice on bulletin board
   4 Memo from personnel office
   1 Verbal request from personnel office
   2 Other, specify:_____

   Supervisor.

   Program Coordinator on site.

2. Did you volunteer to be a mentor?

   8 Yes
   0 No

3. Did the student application provide sufficient information?

   8 Yes
   0 No

4. If no, what additional information would you want to see included on the student application form?

   One mentor commented that questions should be asked concerning technical capabilities.

   I chose a student I was familiar with, I never saw the applications.

5. Did you interview the student who was placed in your laboratory before the program started?

   1 Yes
   7 No

6. If no, would an interview have been useful?

   3 Yes
   3 No

7. In your opinion, how much has the student's work in your laboratory contributed to his/her understanding of the nature of scientific research?

   8 A lot
   0 Some
   0 Not at all

8. How much did the student contribute to the research of your laboratory?

   6 A lot
   2 Some
   0 Not at all

9. How would you rate the student's performance?

   8 Excellent
   0 Fair
   0 Poor

10. Would like to participant as a mentor for the program next summer?

    7 Yes
    1 No
      If No, Why?

    Just every few years.

11. Would you want the same student in your laboratory next summer?

    7 Yes
    1 No
      If No, Why?

    Student starting college.

12. Did the work of the student influence his/her choice of

    a. courses in coming school year? 3 - Yes 1 - No 4 - Don't know
       Explain:

       Most comments were that the courses have already been set. One commented that the student changed from nuclear engineering to electrical engineering.

    b. career choice? 4 - Yes 1 - No 3 - Don't know
       Explain:

       Comments were that the work was parallel to career choices.

If you have suggestions or comments on the program, please use the space below.

PLEASE RETURN BY 14 September 1990

to: Susan Espy
Coordinator

Universal Energy Systems
4401 Dayton-Xenia Road
Dayton, OH 45432
Address

_____
Name of student apprentice

_____
Name of mentor/laboratory

_____
Date

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
APPRENTICE EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A HIGH SCHOOL STUDENT)

(A = A LOT
B = SOME
C = A LITTLE
D = NOT AT ALL)

I. How much were you exposed to each of the following during your summer apprenticeship? (Circle one letter per line.)

A	B	C	D	1.	Philosophy of research
A	B	C	D	2.	Use of scientific method to solve problems
A	B	C	D	3.	Use of experimental checks and controls
A	B	C	D	4.	Measurement techniques
A	B	C	D	5.	Design of equipment
A	B	C	D	6.	Calibration of reagents, standards, and instruments
A	B	C	D	7.	Process of design of an experiment
A	B	C	D	8.	Data analysis (with or without computer assistance)
A	B	C	D	9.	Computer programming
A	B	C	D	10.	Acquisition and use of scientific literature (books, audio visual)
A	B	C	D	11.	Identification of new questions as a consequence of scientific exploration
A	B	C	D	12.	Teamwork in scientific research
A	B	C	D	13.	Use of advanced scientific equipment
A	B	C	D	14.	Other students with similar interests and goals
A	B	C	D	15.	Scientists working in different areas of research
A	B	C	D	16.	Information on scientific careers

II. How much has your experience as an apprentice contributed to your development in each of the following? (Circle one letter per line)

A B C D 1. Working with adults

A B C D 2. Responsibility on a job

A B C D 3. Understanding of scientific principles

A B C D 4. Scientific vocabulary

A B C D 5. Ability to write a technical report

A B C D 6. Understanding of your interests and abilities

A B C D 7. Educational goal setting

A B C D 8. Insights into career opportunities in science

A B C D 9. Career goal setting

(A = A LOT
B = SOME
C = A LITTLE
D = NOT AT ALL
E = NOT AVAILABLE/
NOT RELEVANT)

III. To what extent did you benefit from the following?

A B C D E 1. Planned lectures or seminars

A B C D E 2. Explanations of work by mentor

A B C D E 3. Tours of other laboratories or installations

A B C D E 4. Informal talks with mentor

A B C D E 5. Discussions with other scientists

A B C D E 6. Interactions with other apprentices

A B C D E 7. Advice from the program coordinator

(A = STRONGLY AGREE

B = AGREE
C = DISAGREE
D = STRONGLY DISAGREE)

IV. How do you feel about your research apprentice experience?

A B C D 1. I enjoyed the experience

A B C D 2. I liked the scientific research

A B C D 3. I was satisfied with the way I spent my time

A B C D 4. I learned a lot

A B C D 5. I feel I contributed to the research results

V. Would you want to return to the same mentor next year?

o Yes o No: If No, why?

- o personality conflicts
- o lack of interest
- o want a different experience
- o want a different location

VI. What did you like most about the program?

VII. What did you like least about the program?

DO NOT SIGN

RETURN FORM TO YOUR COORDINATOR BY 14 September 1990
date

Susan Espy
Name of Coordinator

Universal Energy Systems
4401 Dayton-Xenia Rd.
Dayton, OH 45432
Address

1990 USAF/UES HIGH SCHOOL APPRENTICESHIP PROGRAM
APPRENTICE EVALUATION QUESTIONNAIRE
(TO BE COMPLETED BY A HIGH SCHOOL STUDENT)

(A = A LOT
B = SOME
C = A LITTLE
D = NOT AT ALL)

I. How much were you exposed to each of the following during your summer apprenticeship? (Circle one letter per line.)

A	B	C	D		
46	27	26	6	1.	Philosophy of research
36	35	26	8	2.	Use of scientific method to solve problems
45	23	26	11	3.	Use of experimental checks and controls
42	26	18	19	4.	Measurement techniques
38	30	24	13	5.	Design of equipment
30	20	20	35	6.	Calibration of reagents, standards, and instruments
39	31	19	44	7.	Process of design of an experiment
80	17	6	2	8.	Data analysis (with or without computer assistance)
65	17	11	13	9.	Computer programming
47	28	22	8	10.	Acquisition and use of scientific literature (books, audio visual)
34	44	18	9	11.	Identification of new questions as a consequence of scientific exploration
62	30	9	4	12.	Teamwork in scientific research
61	26	11	7	13.	Use of advanced scientific equipment
34	29	26	16	14.	Other students with similar interests and goals
51	29	17	7	15	Scientists working in different areas of research
48	33	20	4	16.	Information on scientific careers

II. How much has your experience as an apprentice contributed to your development in each of the following? (Circle one letter per line)

A	B	C	D		
77	21	7	0	1.	Working with adults
68	27	10	0	2.	Responsibility on a job
46	34	22	3	3.	Understanding of scientific principles
57	27	16	5	4.	Scientific vocabulary
33	46	19	7	5.	Ability to write a technical report
65	30	10	0	6.	Understanding of your interests and abilities
55	33	14	3	7.	Educational goal setting
60	29	13	3	8.	Insights into career opportunities in science
51	33	19	2	9.	Career goal setting

(A = A LOT
B = SOME
C = A LITTLE
D = NOT AT ALL
E = NOT AVAILABLE/
NOT RELEVANT)

III. To what extent did you benefit from the following?

A	B	C	D	E		
15	19	21	2	48	1.	Planned lectures or seminars
73	22	5	4	1	2.	Explanations of work by mentor
27	28	22	5	23	3.	Tours of other laboratories or installations
73	19	7	5	1	4.	Informal talks with mentor
59	27	14	2	3	5.	Discussions with other scientists
36	19	22	10	18	6.	Interactions with other apprentices
16	22	25	18	24	7.	Advice from the program coordinator

(A = STRONGLY AGREE
B = AGREE
C = DISAGREE
D = STRONGLY DISAGREE)

A	B	C	D		
				IV.	How do you feel about your research apprentice experience?
79	22	4	0	1.	I enjoyed the experience
59	37	3	6	2.	I liked the scientific research
51	41	10	3	3.	I was satisfied with the way I spent my time
73	25	6	1	4.	I learned a lot
55	37	5	7	5.	I feel I contributed to the research results

V. Would you want to return to the same mentor next year?

72 Yes 31 No: If No, why?

1	personality conflicts
5	lack of interest
23	want a different experience
10	want a different location

VI. What did you like most about the program?

35 of the students thought that the mentor and the various people in the laboratory was what they liked the most about the program. They commented that they were treated as adults and not as high school students, that what they thought or accomplished was important. 29 of the students liked the exposure to a work atmosphere. The students were very thankful to have the opportunity to work beside scientist and engineers doing real research. Another comment was that the equipment, laboratories, and computers are what 19 of the students liked the most. Eleven of the students liked the project that they were assigned to, and the learning experience that they had. A few students expressed that before they received the position that they had not decided on a career, but 8 students have now due to the program. Five of the students liked the different employment opportunities that are available, while 2 of the students liked learning more about the Air Force and the opportunities that they have available.

VII. What did you like least about the program?

The majority of what the students liked least was that the program was not long enough. The 12 responses indicated that the program should be lengthened to 10 to 12 weeks. Eleven of the students commented at they were not keep busy. The mentors did not either have the time to spend with them, or the students finished the projects that the mentors had assigned. Six of the students disliked writing a final report at the end of the summer. While another six students thought the pay should be higher considering the type of work they were doing. Five of the students commented that they did not do the project that was originally discussed once they got there. Another five students least liked the timecards and there schedule, also the delay in getting their paychecks. Four students responded that their mentors were TDY and was not available for most of the summer. Four other students commented that they did not get along with the people in labs, that they were treated like "gofers" and doing errands, and office work. Another 4 students disliked the project they were doing, some of the comments were that they did not think it was real research but "busy work". Other comments consisted of no sick or holiday pay, lack of information about UES, no positions available for college students, getting up early, lack of air conditioning, and even that the water tasted funny.

DO NOT SIGN

RETURN FORM TO YOUR COORDINATOR BY 14 September 1990
date

Susan Espy
Name of Coordinator

Universal Energy Systems
4401 Dayton-Xenia Rd.
Dayton, OH 45432
Address
2361s

LIST OF PARTICIPANTS FINAL REPORTS

RESEARCH REPORTS

1990 HIGH SCHOOL APPRENTICESHIP PROGRAM

Technical Report Number	Title	Participant
VOLUME I		
Aero Propulsion Laboratory		
1	Flash Plate Evaporator	Matthew Bold
2	Frozen Start Up	Hee Sun Choung
3	Nonaqueous Battery Research	Katharine Day
4	Setup Tecplot	Chris Hatch
5	Flash Plate Evaporator	Chet Nieter
6	Final Report to UES	Jennifer Pollock
7	Frozen Start Up	Carol Rogers
Armament Laboratory		
8	Wind Velocimeters for Calculating Ballistic Trajectories	Steven Bryan
9	Star Availability for Sensor - Specific Evaluation (S.A.S.S.E.)	Tonya Cook
10	Development of a Customized Database System for the Distribution of EPIC Hydrocode Software	Heather Cox
11	Synthesis and Characterization of 3-Picrylamino-1,2,4-Triazole	Kathryn Deibler
12	Neural Target Identification	Chris Ellis
13	Space Debris Analysis	Dana Farver
14	Enhancement and Integration of Post Process Utilities for the EPIC Hydrocodes	Kenneth Gage
15	Design of In-House Radar Control and Data Acquisition Systems	Reid Harrison
16	1990 HSAP Final Event Summary	Derek Holland

17	MSIS: Multi-Sensor Integration System	Christine Riendeau
18	Enhancement of RTD 710A Interface	Lisa Schmidt
19	Current Simulations in Electromagnetic Launcher Power Supplies	Patricia Tu
20	Advanced Signal Processing Operations for Guided Interceptors	Troy Urquhart
21	Ballistics Applications in Aerospace Research	Greg VanWiggeren
22	Optical Processing: Digital Imagery Acquisition and Analysis	Danielle Walker
23	Fractal Landscape	Eric White

VOLUME II
Armstrong Aerospace Medical Research Laboratory

24	Summer Apprenticeship Final Report	Rex Ballinger
25	3-D Audio Displays	Douglas Brungart
26	Integrated Protection by Pressurized Containment in Flight Environments	Caroline Chuang
27	Cockpit Accommodation	Keisha Hayes
28	Electron Microscopy Techniques	Douglas Marshak
29	AAMRL/AFOSR Summer Research Apprenticeship Report	Jeremiah Rogers
30	Camouflage, Concealment, and Deception	James Shamiyeh

Arnold Engineering and Development Center

31	Converting Saturn Data Base into Paradox 3.0	Timothy Craddock
32	X-ray Computer Tomography and IR Analysis Models for Propulsion Systems	Myra Medley
33	Demulsification of Oil and Water Using Salts and Commercial Surfactants for the Purpose Purpose of Reclaiming Waste Oil	Julie Reece
34	Propulsion Wind Tunnel Standard Tunnel Conditions Equations Documentation	Jonathan Sanders

35	Program Final Report	Jason Scott
36	Real-Time Radiography Ray Tracing	Gerald Turner
Astronautics Laboratory		
37	The Summer in Review	Alisha Conrow
38	Synthesis of Two Cage Compounds	Debra Meyer
39	Fundamental Rocket Exhaust Measurements	John Moro
40	Analytic Predictions of Hydroxy-terminated Polybutadiene (HTPB) Modulus using Random Iterative Discrete Node Algorithm (RIDNA)	Lloyd Neurauter
41	Liquid Engine Computer Codes	Joseph Padilla
42	Determination of Active Surface Area and Density of Carbons	Melanie Pyle
43	No Report Submitted	Thomas Quinn
44	High School Apprenticeship Program Final Report	Tracy Reed
45	1990 Final Report	Benjamin Sommers
46	No Report Submitted	Stephanie VanMeter
47	A Comparison of Electric Propulsion Orbit Transfer Methods	Rebecca Weston
48	No Report Submitted	David Youmans
Avionics Laboratory		
49	Ada Compiler Evaluation Capability Testing Utility (ACEC)	Brian Barclay
50	Pattern Based Machine Learning	Mark Boeke
51	Pattern Based Machine Learning: A Comparison of ADA Function Decomposer Versions	Michael Chabinyc
52	Computer Simulation of SDI	David Collins
53	Final Report Summer 1990	Austin Flack
54	No Report Submitted	Jerard Wilson

Engineering and Services Center

55	Surface Catalyzed Reactions of Vapor Phase Hydrazine	Jennifer Brewer
56	Sims. Billeting	Philip Dorsch
57	Alternate Pavement Materials	David Eshleman
58	Apprenticeship Final Report	Richard Hartzer
59	A Neural Network Edge Enhancer	Thor Johnson
60	Summer Research	Tracy Lamb
61	Final Report	Brent Miller
62	Effects of Compaction on Unsaturated Sand	Debra Piechowiak
63	Design and Testing of an In-house Electronic Publishing System for National Laboratories	Jonathan Protz
64	Summer Apprenticeship at HQ AFESC/DEC	Julie Scruggs
65	No Report Submitted	Michael Stone
66	The Effects of High Pressure Tires	Amy Thomas
67	The Effects of High Strain Rates on Sand	Michael Thompson
68	Centrifuge Modeling of Explosive Induced Stress Waves in Unsaturated Sand	Jeffrey Ward
69	CD-ROM Compact Disc Read Only Memory	Robin Woodworth

VOLUME III
Electronic Technology Laboratory

70	Final Report	Matthew Brewer
71	The Making of a Transistor	Matt Elwood
72	Imagepro User Guide	Shelly Knupp
73	Computer-Aided Design (CAD) Area	Christopher O'Dell
74	Electron Beam Lithography	Suzette Yu

Flight Dynamics Laboratory

75	High School Apprenticeship Program Final Report	Jean Ay
76	Project "Environmental Reliability": The Analysis of Printed Circuit boards	Matthew Becker
77	High Speed Performance Computer Resources Team	Wendy Choate
78	No Report Submitted	Andrea Dean
79	Leading Edge Heat Exchanger	Rachael Lyon
80	Final Report for Cathie Moore	Cathie Moore
81	Project Instrumentation	Roderick Morgan
82	ENTRAN Manual	Stanley Wall
83	Final Report	Douglas Wickert

Geophysics Laboratory

84	Determining Tropical Storm Direction of Movement Using SSM/I Brightness Temperature Data	Stephen Britten
85	Auroral Boundaries	Weihaw Chuang
86	Mesoscale Modeling	Christopher Guild
87	No Report Submitted	Jason Klingensmith
88	Solar Terrestrial Interactions	Galen McKinley
89	Ionospheric Effects	Jeffrey Sayasane
90	Final Report	Paul Swietek

Materials Laboratory

91	Fatigue of Composites	Jennifer Walker

Occupational and Environmental Health Laboratory

92	Automated Heat Stress Measurements	Gary New
93	Compilation of Hazardous Waste Surveys	Andrea Dean (Perez)
94	Automation of Water Analysis	Michael Smid

Rome Air Development Center

95	Pre-Commissioning and Capabilities of the Workhorse Scatterometer: A Final Report	Da l Abbis
96	No Report Submitted	Mark Anania
97	Radar Range Equation and Radar Design and Simulation	Bridget Bordiuk
98	No Report Submitted	Todd Gleason
99	Cartographic Applications	Christopher Hailes
100	Application Software Development for use on Directory Specific Network	Edward Holmes
101	Various Projects in Data Analysis of Surveillance Systems	Kimberly King
102	Network Processing Element Test and Evaluation	Kathryn Lee
103	Gemacs Version 5.0 Modifications and Applications	Jason Lenio
104	Activities at RADC Photonics Lab	Kevin Olson
105	Overview of Neural Networks	David Petrillo
106	Surface Analysis	Thomas Potter
107	Software Inventory Control System Database	Daniel Russell
108	Artificial Neural Networks	Philip Schremmer
109	Local Area Networks	Eric Shaw

VOLUME IV
School of Aerospace Medicine

110	No Report Submitted	Anthony Barnes
111	Final Report	Whitney Brandt
112	Preliminary Testing of an Inert Gas Concentrator Using Carbon Molecular Sieve	Deann Cooper
113	Affects of Anti-G Straining Maneuvers on Blood Pressures in Man; Affects of +Gz on the Cardiovascular System of Baboons	Matthew Felder

114	Apolipoprotein Comparison Study and CAD Prediction Study	Christopher Hudson
115	High School Apprenticeship Program Final Report	Soyna Longbotham
116	Determination of Optimum Growth and Strain Facilities for Diazotyrosine Production	Brian McBurnett
117	Serum Factor - Induced Expression of the C-FOS Gene in NIH 3T3 Cells	Heather Neville
118	Biological Rhythms Research	Lori Olenick
119	Report on File with UES & SAM	Joanna Saucedo
120	Medical Entomology Activities	Wendy Shields
121	Establishing High and Low Rate Operant Behavior in Rats	Brent Strawn
122	UES Report	John Taboada

Weapons Laboratory

123	Phase Conjugation and Beam Coupling	David Cochrell
124	Final Report	Paul Hillman
125	A Study in C	David Knapp
126	Final Report of Summer Mentor	Aaron Laing
127	Statistical Analysis of Cone Penetrometer Data	Kerim Martinez
128	No Report Submitted	Ryan McAlhaney
129	Microwave Susceptibility Testing of Micro-Circuitry	Margaret Morecock
130	Construction of a Delay Circuit for Data Acquisition Coordination	Philip Ortiz
131	Contour Plots	Brian Rizzoli
132	Wide Bandwidth Crosslink Register Technical Report	Chris Stoltenberg

FINAL REPORTS

AERO PROPULSION LABORATORY

APPRENTICESHIP PROGRAM
WRIGHT PATTERSON AIR FORCE BASE
WRIGHT RESEARCH & DEVELOPMENT CENTER

MATT BOLD
cowritten with Chet Nieter
MENTOR: DR. WON SOON CHANG

PROJECT: FLASH PLATE EVAPORATOR

The heat pipe was designed in the 1940's to advance the science of heat transfer. The heat pipe has three primary sections: the evaporator, the adiabatic section, and the condenser section. A working fluid in the evaporator is vaporized by incoming thermal energy. The vapor pressure increases and the vapor is forced through the adiabatic section to the condenser where the vapor is condensed and thermal energy is released. The condensed fluid is drawn back to the evaporator section through a porous wick structure. This is made possible by a drop in capillary pressure in the evaporator section of the pipe. In essence, the thermal energy was transferred from the evaporator section of the pipe to the condenser.

Normally the heat pipe is in the form of a long cylindrical pipe used to transfer heat over long distances. A new type of pipe was needed to enhance the ability to collect and remove heat over large surfaces, or to increase the radiation area.

The flash plate evaporator is the third unit in a three unit system designed to remove large quantities of thermal energy from tightly packed electronics primarily on board spacecraft. The flat plate heat pipe, and the side flow heat pipe, the first two units were tested last year. The flat plate removes the thermal energy from the electronic units to be cooled and the side flow transfers the energy to the flash plate where it is radiated away from the craft.

The performance of the flash plate was the primary concern of our research this year. Its effects of both forced and free convection were studied. Effects of gravity and the loss of energy

through the insulation and edge effects were also a concern.

The plate's facing was fabricated form aluminum as was the wick and honeycomb structure inside. The working fluid was acetone. The plate has the dimensions of 11 3/8" x 11 5/8" with a thickness of .36".

The first tests run on the flash plate were free convection tests. The flash plate was heated by 11" x 3" foil resistance heater. Temperature readings were taken every five minutes from eleven thermocouples placed on the flash plate to a Fluke 2286A Datalogger. Typical tests involved starting the heater at 30 watts and letting the flash plate reach steady state, then increasing the power another 10 watts and again allowing the plate to reach steady state. A maximum power test was run to see how much power the plate would transfer at its safe operating temperature of 75 degrees C, this was found to be 80 watts. Tests were also run while tilting the plate at various angles to determine the effects of gravity on the plate's operation; tilts of 90 degrees, 45 degrees, 30 degrees, and 15 degrees were used. The temperature readings were then plotted into graphs using Lotus 123. Using these graphs, irregularities were looked for in the data (fig a).

The set up for the forced convection tests were different. A 11 1/2" x 11 1/2" copper cool block was placed on the condenser side of the plate. A refrigeration and circulation unit was used to circulate SR-1 Dowtherm heat exchange fluid. Six thermocouples were attached to the evaporator side of the flash plate, a thermocouple probe was placed to read the temperature of the cool unit, the temperature of the flow into the cool block, and the

temperature of the flow out of the cool block. Also connected to the Fluke was a flowmeter. As with the free convection, the temperature readings were graphed using Lotus (fig b).

The amount of heat lost during the free convection test was calculated using the following equations.

$$Q=hA(\Delta T)$$

The "h" is found using the following equation;

$$h=\frac{Nu_l K}{L}$$

"Nu" is found using;

$$Nu_l=[0.825+\frac{0.387Re_l^{1/6}}{[1+(0.4121\times Pr)^{9/16}]^{8/27}}]^2$$

and "Re" is found using;

$$Re_l=\frac{g\beta(\Delta T)L^3}{V\alpha}$$

It was found that there was a small loss of energy probably due to edge effects and loss back through the insulation.

Two other projects were worked on during the summer. The study of the effects of magnetism on the capillary pressure of liquid oxygen was started. We assisted in the construction of a condenser for the liquid oxygen. Because of the volatility of liquid oxygen it was not feasible to keep large quantities of the substance in the lab.

We also attempted to construct a pipe using unidirectional nickel wick. A square pipe was first constructed and work was begun on a method for creating a round pipe.

Further development of the flash plate and the flat plate system will allow spacecraft to carry more electronics or more advanced electronics that produce larger quantities of thermal energy. Liquid oxygen working fluids could be assisted by a magnetic field increasing the capacity or the pipe that it is used in. The nickel wick can be used in advanced diode heat pipes in the future. More research must still be done but all of the goals are feasible.



BIBLIOGRAPHY

Chi, S.W. Heat Pipe Theory and Practice, New York: Mcgraw-Hill Book Company, 1976, pp.1-94.

Dunn, P.D. and Reay, D.A. Heat Pipes, Oxford: Pergamon press, 1982, pp. 1-10.

Fleischman, G.L. High Power Spacecraft Thermal Management, Hughes Aircraft Company, electron dynamics division, 1987, pp.55-61.

ACKNOWLEDGEMENTS

The work done was sponsored by the Aero Propulsion and Power Laboratory at the Air Force Wright Research and Development Center. I would like to show my appreciation to the following people: Dr. Won Soon Chang, Brian Hager, Mike Ryan, Don Brigner, John Tennant, Don Reinmuller, and Joella Pinckney.

Condenser Side

FLASH PLATE EVAP

JUL 17 TEST NO.FL00130

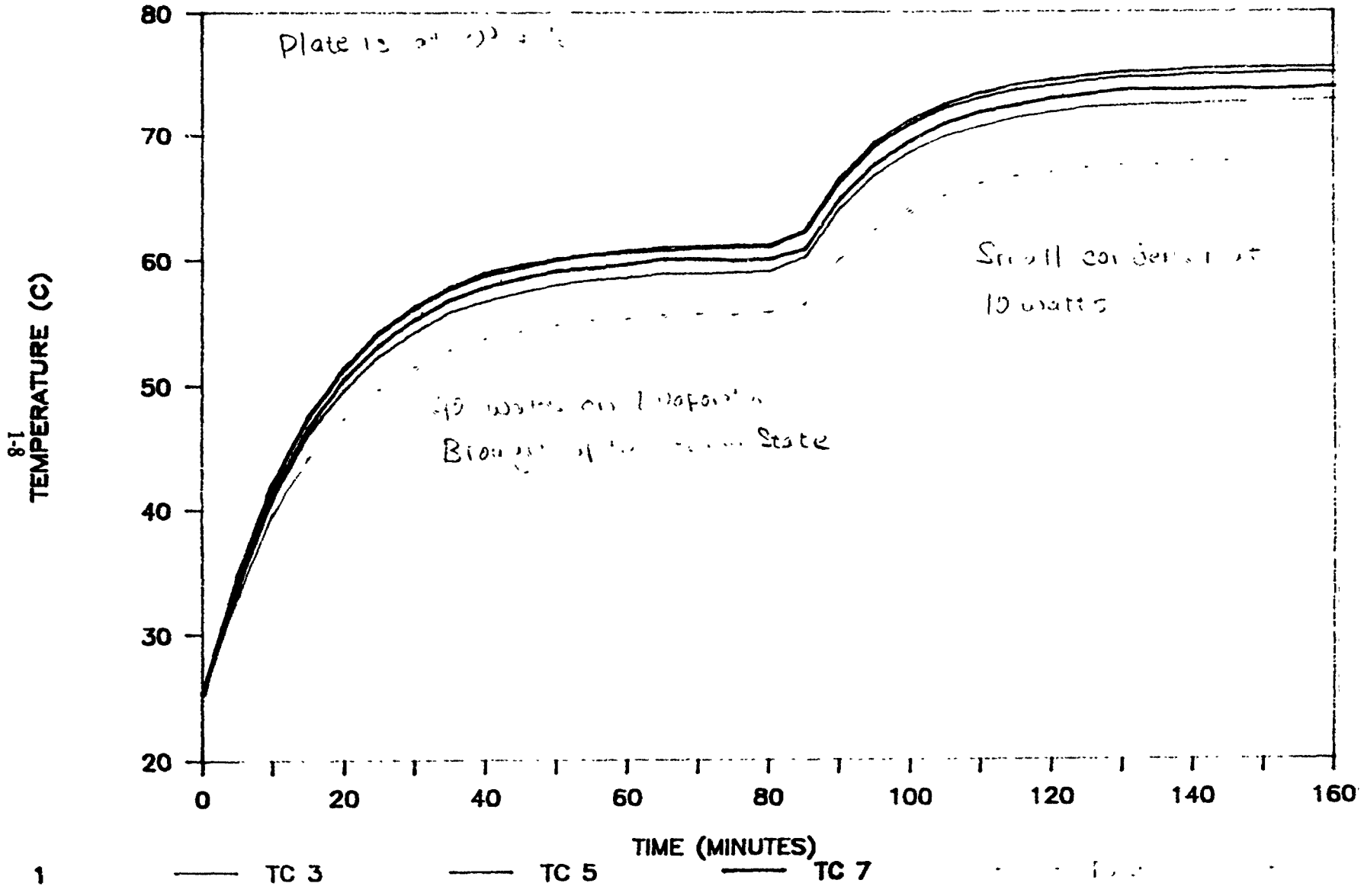


fig a

FLASH PLATE EVAP

AUG 8 TEST NO.FL00180

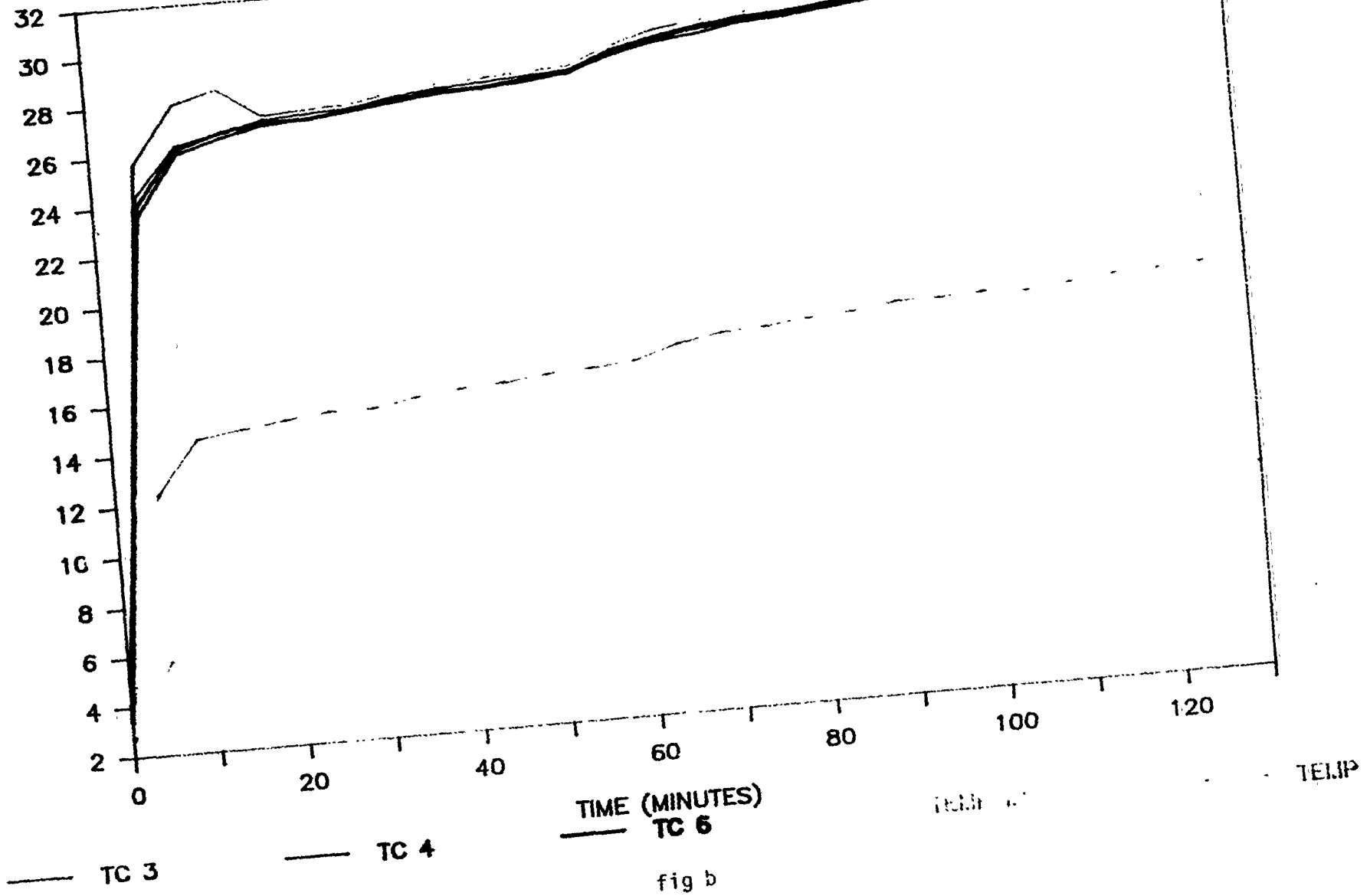


fig b

"FROZEN START UP"

HEESUN CHOUNG
MENTOR: BRIAN G. HAGER

AUGUST 13, 1990

In conjunction with the Air Force High School Apprenticeship Program and Universal Energy Systems, I was assigned to the project, "Frozen Start Up" which was being conducted in the thermal division of the Aero Propulsion Laboratory. "Frozen Start Up" is an experiment which involves heat pipes.

Heat pipes are designed to transport large amounts of heat from one location to another over a small temperature gradient with a small unit size. The design and operating principles of the heat pipe are simple. It is a closed tube or chamber whose inner surface is lined with a porous capillary wick. The wick is saturated with a liquid working fluid. Heat is then applied to the evaporator end of the pipe, thus allowing the liquid to vaporize. The vapor travels through the vapor core to the condenser section where it condenses releasing the latent heat of vaporization. A capillary pressure is created by the depletion of liquid in the wick, causing the liquid to travel from the condenser to the evaporator. The heat pipe should continually transport the latent heat of vaporization from the evaporator to the condenser without drying out the wick.

One practical application of the heat pipe is for thermal control in spacecraft. Realizing that in space the heat pipe must operate in sub-zero temperatures, research is currently being done in the frozen state to simulate the conditions in space. However, there are several problems associated with frozen heat pipes. The most serious of these is frozen start up. A frozen pipe usually dries out before it becomes fully operational. The solid liquifies and eventually vaporizes in the evaporator section of the pipe.

However, the working fluid often remains in its solid state throughout the middle and condenser section preventing the condensed vapor from traveling up to the evaporator. Thus, dry out occurs.

The purpose for this "frozen start up" experiment is to add to the general research data available. Similar tests have been conducted by W. Bowman of the United States Air Force Academy providing results showing a successful "frozen start up."

The first of the procedures followed were calculations. These calculations were used to determine the point at which the pipe would dry out. Taking into account the characteristics of the pipe, the information in the following equations were used to find the capillary, sonic, boiling, and entrainment limits, respectively.

$$Q_{c_{max}} = \frac{(QL)_{c_{max}}}{(0.5L_c+L_a+0.5L_e)}$$

$$Q_{s_{max}} = A_v \rho_v \lambda \sqrt{\left[\frac{\gamma_v R_v T_v}{2(\gamma_v+1)}\right]}$$

$$Q_{b_{max}} = \frac{2\pi L_e K_e T_v}{\lambda \rho_v \ln(r_1/r_v)}\left(\frac{2\sigma}{r_n}-P_c\right)$$

$$Q_{e_{max}} = A_v \lambda \sqrt{\left(\frac{\sigma \rho_v}{2 r_{h_s}}\right)}$$

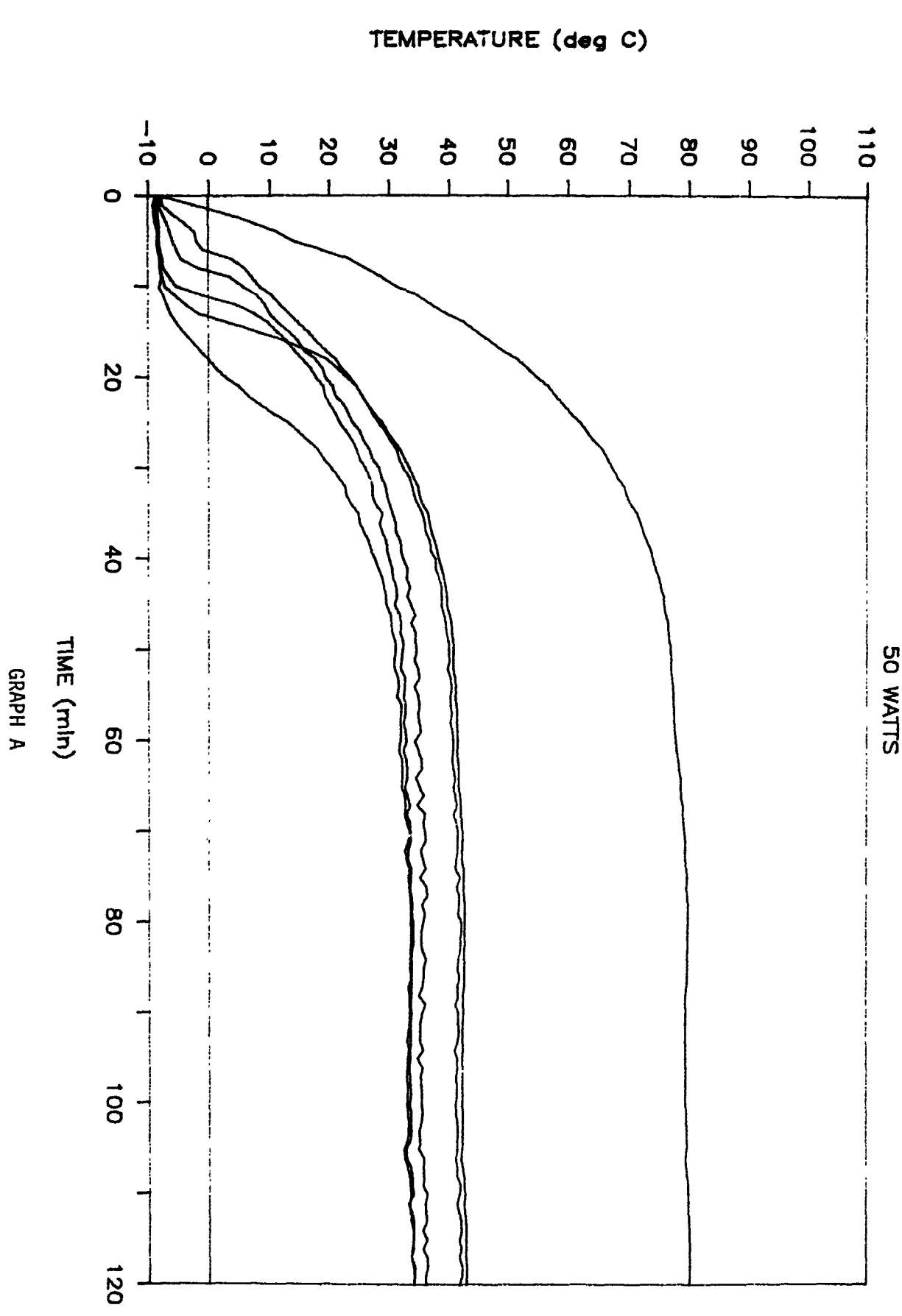
The next step involved the instrumentation of the pipe. The copper pipe was 24 inches long and had a diameter of one-half inch. The wick inside the pipe was a copper wire screen with the working fluid being water. Type T thermocouples were placed every two inches along the pipe in order to read the temperatures of the pipe. These thermocouples were then connected to a data logging system, Fluke 2286A. After programming the Fluke to record temperatures at given time intervals, tests were started. The first tests were characterizations with different amounts of power supplied to the heat source. The heat source was a clam shell heater capable of reaching temperatures above 600 degrees Celsius. Characterization tests were run to provide data in the form of graphs for comparison to "frozen start up" tests.

When all the characterizations of the pipe were completed, "frozen start up" tests were implemented. Cooling the environmental chamber at -15 degrees Celsius, 50 watts and 100 watts of power were added to the heater to begin the process of "frozen start up." Other tests were conducted with the chamber at -25 degrees Celsius and -65 degrees Celsius. After analyzing the results of the tests, it became apparent that the heater was not generating the proper amount of heat for its wattage due to the fact that most of its power was put into heating its own mass. Therefore, it was decided to replace the clam shell heater with a

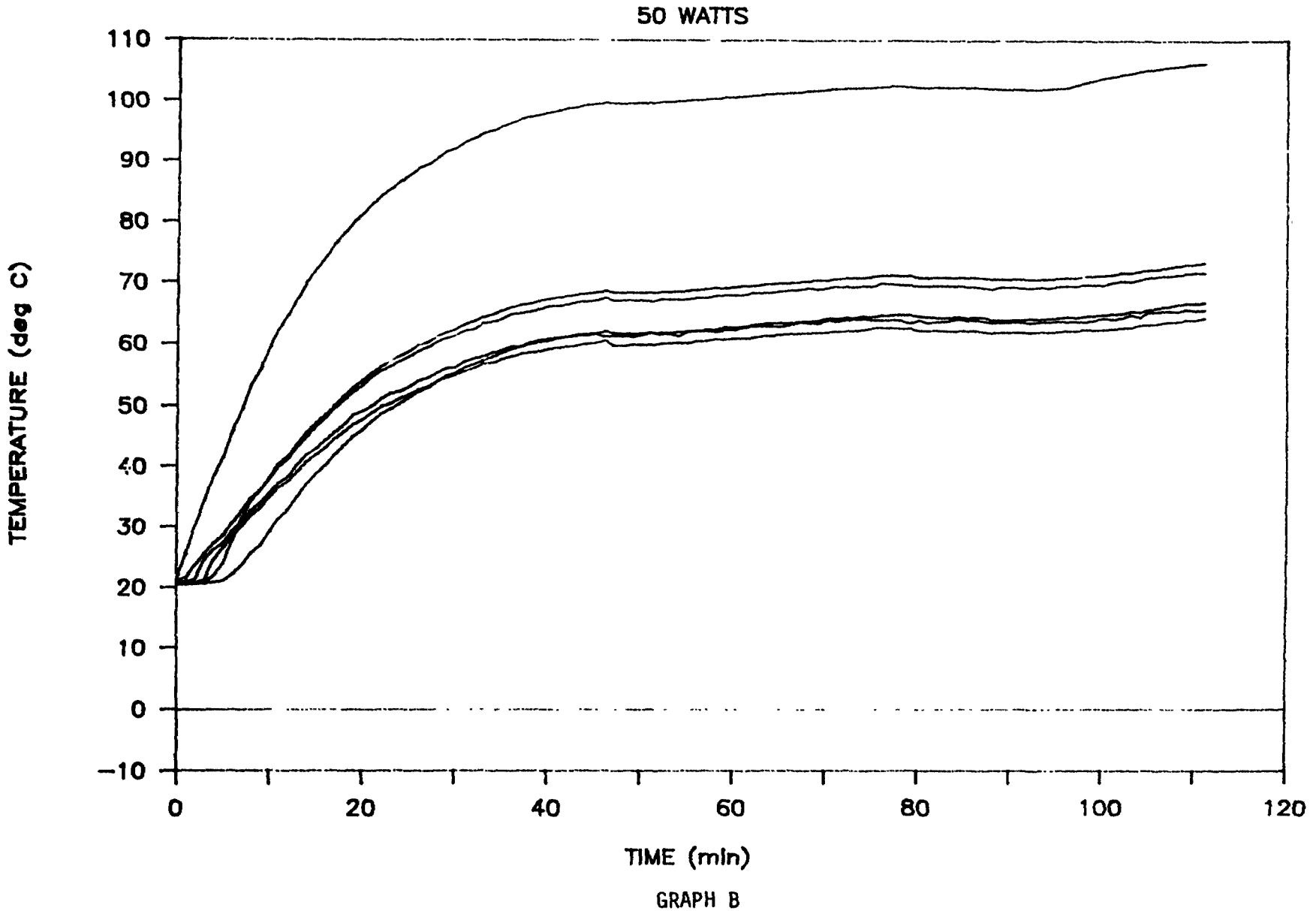
smaller, more efficient one. Again, characterizations were run for comparison using the smaller heater.

With a new temperature controller, the chamber was cooled to a constant -10 degree Celsius. To the frozen pipe, different amounts of power were added to conduct "frozen start up" tests at 50 watts and 100 watts. By comparing the graphs of these tests to the characterizations, it was concluded that successful "frozen start ups" have been achieved (See graph A and B).

This experiment has provided valuable data for frozen heat pipes. Further tests could be done with different working fluids and/or wick types. This will eventually make heat pipes vital to science and technology in space.


TEMPERATURE (deg C)
-10
0
10
20
30
40
50
60
70
80
90
100
110
0
20
40
60
80
100
120
TIME (min)
FSU AT -10 DEG C
50 WATTS
GRAPH A

CHARACTERIZATION OF HEAT PIPE



GRAPH B

BIBLIOGRAPHY

Chi, S.W. Heat Pipe Theory and Practice, New York: Mcgraw-Hill Book Company, 1976, pp. 1-94.

Dunn, P.D. and Reay, D.A. Heat Pipes, Oxford: Pergamon Press, 1982, pp. 1-10.

Acknowledgements

This research was sponsored by the Aero Propulsion and Power Laboratory at the Air Force Wright Research and Development Center. I would like to thank the following people for their guidance and assistance with the "Frozen Start Up" project: Brian Hager, Dr. Won Soon Chang, Mike Ryan, Don Brigner, John Tennant, Joe Mantle, and JoElla Pinckney.

NONAQUEOUS BATTERY RESEARCH

by Katharine M. Day

Mentor: Richard Marsh
Aero Propulsion and
Power Laboratory -
Battery Lab
8/22/90

ACKNOWLEDGMENTS

I would like to thank my mentor, Richard Marsh, for getting me settled in and introducing me to my co-workers. Mostly, however, I would like to offer my thanks and appreciation to the two men with whom I spent my apprenticeship, Allen Turner and David Ryan. They took time out of their busy schedules to include me in their research. I was given their full respect and confidence which allowed me to work well with them. Finally, I would like to thank all the employees of the Battery Lab for receiving me warmly and treating me so kindly. These people include: Scott Bishop, Sue Tanner, Steve Vukson, Stephan Wolanczyk, Gary Loeber, Joe Fellnerr, Mel French, Richard Flake, and Cam Riepenhoff. Thanks to you all for a wonderful experience.

I. INTRODUCTION

At the time of my apprenticeship, the project "Lithium Batteries" was inactive, therefore, I was assigned to work on a project entitled "Nonaqueous Battery Research." This "in-house" project is being carried out in conjunction with a contractor at Wright State University and has been active for less than a year. The project, as its name implies, deals not with aqueous electrolyte batteries but with molten electrolyte batteries. Unfortunately, I was unable to be involved in the actual testing of the cells, however, I took part in the before and after stages of the research and learned many things.

II. GENERAL DESCRIPTION OF RESEARCH

During my apprenticeship, I was exposed to the background and history of the the project as well as the current research. For instance, the process begins when David Ryan chooses a compound that he believes will yield a cell with a high energy density. This compound forms the cathode of the cell. Next, the technician synthesizes the electrolyte in the laboratory by mixing lithium chloride and potassium chloride at their eutectic point. The eutectic point is the quantitative mixture of the two chemicals that has the lowest possible melting point.

However, a binder must be added to keep the electrolyte from flowing in its molten state. A lithium-aluminum alloy mixed with a percentage of electrolyte composes the anode. These parts are made into a cell and then tested. The results are compared with existing standards to check the progress of the project. The goals of this research project include either increasing power density over existing materials or keeping the same power and making the cells smaller and lighter in order to save space and money. The development and perfection of these cells will then allow for various uses in industry. For instance, these particular cells are thermal batteries which are used in such things as weapons, aircraft, missles (i.e. tow-missles), and bombs. Basically, they are employed where electric power is needed for a short duration with a long storage life and quick access. Therefore, this research project fits into the larger space and military research done by the laboratory. By observing the specific plans for the use of thermal batteries, it becomes obvious that Wright Patterson Air Force Base is in a position to make a great contribution to the field.

III. DETAILED DESCRIPTION OF RESEARCH

Although the basics of the project may seem simple, the preparation, testing, and compiling of results involves

a long and complicated process. The most difficult and tedious task is the synthesis of the electrolyte. First, the lithium chloride and potassium chloride are measured out and mixed together in a twin-shell blender. The mixture is then placed in a oven at 500°C and melted into a clear liquid. After cooling, the next step is grinding the mixture into a very fine powder. Then 10% by weight of the binder is added. The salt is placed in the bottom of a special type of dish with the binder on the top, and then the dish is placed in an oven. At 400°C the salt melts, but the temperature must be 500 to 525°C before the binder can be mixed in. The process involves stirring for a while then letting it heat back up. These two activities are alternated for most of an 8 hour work day until the mixture is uniform. Then it is cooled until it becomes brittle, and finally it is ground into an extremely fine powder. This entire process takes place behind the doors of a Dry Room. This provides a cool, moisture-free atmosphere for working with certain chemicals. Thus, the electrolyte is completed and set aside for the time being.

After the engineer selects a compound for the cathode material, the contractor at Wright State University synthesizes it in powder form. Then pellets of each anode, cathode, and electrolyte material are made. The technician places 1.5 grams of anode material (lithium-aluminum alloy plus electrolyte), 3 grams of cathode material, and 3 grams

of electrolyte individually in a hydraulic press at 50,000 pounds of pressure. The resulting pellets have a 2 inch diameter and are very thin. These pellets are stacked with the electrolyte in the middle to form cells. Nickel "dumbbells" or connectors are made by spot welding nickel strips to nickel circles that have been cut with a punch and die. These connectors are taped onto either end of the cell. Holes are punched in squares of fibrofrax large enough to fit the cells which are then taped to the fibrofrax, making sure to seperate the negative connectors from the positive connectors. Next, a piece of mica is placed on either side of the cell and taped together. This mica provides insulation so that the cell does not get too hot during testing.

The testing is done on a single cell tester in a dry box with an atmosphere of argon. The cell is placed between two plates while a computer puts a load on the cell. The load is chosen by the operator and can be changed to show various effects. The cell is heated up and allowed to run for 60 seconds in order to get a peak voltage reading, and then the load is removed for 1 second and replaced. After that, the load is removed at 30 second intervals for 1 second until the cell discharges completely. The data (current and voltage) are recorded on a plotter and then analyzed. The 70, 80, and 90% lifetimes are measured, and the power density is calculated in

watt-hours per kilogram and watt-hours per pound. These results are then compared to the power density measured from the standard cells that use FeS_2 as their cathode material. The cells are then dismantled and given a "post-mortem" examination. This is to record any physical characteristics such as color, texture, or hardness that might tell of changes and/or problems that occur during testing. This step ends the long, complicated process of testing nonaqueous thermal batteries.

V. RESULTS

The results of the project are not nearly as confusing as the procedure. Basically, two vanadium oxide compounds have been tested to date. Out of these two, both have failed to produce higher energy densities than the standard FeS_2 cells. The next compound, a sodium, vanadium oxide compound will be tested as soon as the computer program is modified. The engineer does not think that this one will be any better than the two previous compounds. However, he says that he has another compound presently being made that will be a good cathode material. Therefore, so far the project has been basically unsuccessful, but it has a bright future that is sure to yield a worthy compound.

VI. OTHER INTERESTING OBSERVATIONS AND LESSONS LEARNED

Due to the pause in research caused by the modification of the computer program and the single cell tester, I had the opportunity to work on another "project." The Environmental Protection Agency is beginning to crack down on the careless habits of the government scientists, therefore, a special program must be followed by the laboratory. I was very involved in the reorganization of the lab to fit the guidelines of this new program. I helped take inventory of all the chemicals owned by the lab and then I looked up codes and labels for each of them in Federal Codes and Regulations books. After that I made appropriate labels and helped seperate and rearrange all of the chemicals on the proper shelves. Finally, I helped take another inventory and posted a list of chemicals at each storage location. I was also in charge of making sure that there was a Material Safety Data Sheet in a set of books in 5 different locations for every chemical in stock. The last step, however, was beginning to fill out "2005 forms" for the removal of all of the chemicals in the basement storage area. I also worked on several smaller projects such as designing a new part for the single cell tester and plumbing an apparatus for the hook up of gas to an instrument. Therefore, during my apprenticeship, I was always busy and learning many new things. Overall, I

believe that the experience was well worth all of the hard work.

BIBLIOGRAPHY

None used.

Final Report

by

Chris Hatch

for

Summer Mentorship Program
Aug 16, 1990

I met Greg Cala through a mentorship program at school. Since we got along well, he requested me for this program. I want to thank him for this.

I was already somewhat familiar with the layout of the lab, so the first few days I spent getting to know the computer setup being used. Mostly I had to use Microsoft DOS, Microsoft Quickbasic, HP Basic, and Unix, but I learned some Fortran to aid my understanding of an already existing code.

My first assignment was to setup Tecplot (a graphing program) so that Mr. Cala could take data from the math model of a planned turbine testing facility and obtain meaningful graphic representations of it. The math model was running on the local Vax, so Mr. Cala and I set up the PC so that we could import files from the Vax through an Iris workstation.

The next thing I had to do was set up ten different stylesheets for the ten different sets of data that the math model produced. It was necessary to make two stylesheets for each dataset one for output on the TI Laserwriter and the other for output on a color laser printer.

The stylesheets on Tecplot didn't contain all the necessary information for the printouts Mr. Cala wanted, so I also set up batch files controlling the default settings of Tecplot. To aid the processing of data from the math model, I created a playback file on Tecplot that automatically loaded, manipulated and printed all ten datasets created by a single run of the math model.

Once I had finished setting up Tecplot on the PC (a Northgate 386) I started the same process on an Iris workstation. This required that I learn the Unix operating system, so I spent a day doing only that.

The Iris was running an earlier version of Tecplot and was not fully compatible with the PC version. This meant that I had to redo much of my earlier work on the PC. I also created a playback file on the Iris that loaded and displayed every dataset from a math model run on the monitor at once. This was an easy and fast way of scanning data as it was produced, but the PC had been too slow to run this application.

Mr. Cala's main project at this time seemed to be overseeing the construction of ATARR (Advanced Turbine Aerothermal Research Rig), a turbine testing facility with a run time of just under a second. The math model that I set Tecplot up for was of this facility.

Over the next couple days Mr. Cala had me catalog the lab's supply of hotwires using D-Base III. The lab had almost one hundred hotwires, many of them as old as I am, and they had never been properly inventoried.

I then began to trace and properly label the wiring between the thermocouples on the test table in the hanger bay outside and the HP Voltmeter inside the lab. The test table had been used by several graduate students, each of whom had changed it, and there was no useful documentation on which thermocouples were connected to which channels on the voltmeter, or even whether or not particular thermocouples were working.

This took several days, as I had to trace every wire by hand,

check every thermocouple for continuity, and run tests at different temperatures to see which thermocouples still functioned properly.

Once I had finished this I organized my data, made circuit diagrams of the entire setup, and put all the data into the PC. I used, because of the computer setup in the lab, Lotus 123 version 2.0 and 3.0, Quattro Pro, and D-Base _II. I had it so that changes to the thermocouple setup could be entered in either Lotus or Quattro, where data manipulation was easiest, and then transferred to D-Base so that the data could be sorted and printed.

The second to last week I worked at Wright Patterson Mr. Cala went on vacation gave me several projects to work on while he was gone. First I was to transfer a floppy disk containing inventory information for all the equipment in the lab to a spreadsheet. This proved difficult because of the extreme size and length of the file and because the fields in it were not properly delimited. I failed to fit this file into a spreadsheet, but when Mr. Cala came back, we succeeded in doing so by splitting the inventory file up into several blocks and compressing the data on it.

Another project that Mr. Cala left me was tracing the menus of the Acquire software that he uses for data acquisition. I rapidly finished this and started familiarizing myself with HP Basic.

The reason the lab still used eight year old HP's was that all the data acquisition software we had was written for this machine. Mr. Cala had written a utility program in HP Basic that allowed for the automatic copying or purging of several files on any drive in the machine. He wanted me to modify this code to do several different things. Most importantly, he wanted to ability to keep the old date on a file when it was moved or copied. I had some trouble with this since it was not something that the creators of HP Basic had thought of doing and the manuals were vague.

The other main addition I was to make to his code was the ability to scan a section of hard disk to locate and then remove all unwritable gaps in the disk. These gaps were created because the HP does not fragment files like an IBM. When a file is deleted on the HP, a hole in both the disk and the directory is created. The hole has the size of the file that was stored in it, but it is represented by only one hole in the directory. If a file is read into a hole larger than it is, the hole in directory is filled, even though the hole on the disk has not been completely filled. Because of this, one five meg. section of hard disk had two meg. of inaccessible holes. I wrote compression routines that recovered this wasted space. To make the process as fast as possible, I used a ramdisk for some functions. Unfortunately, the HP has only four meg. of memory and sometimes my program corrupts this memory, making it necessary to reboot the system after the compression routine has finished.

My last assignment was to evaluate Graftool for Mr. McAurther. He was trying to find a suitable graphing program that the lab could accept as a standard. Since I was the most familiar with Tecplot, I was qualified to compare it with Graftool. The programs varied widely and I was not able to solidly recommend one or the other, so my report simply pointed out the capabilities of each

program and the applications that they were the best suited for.

Mr. Cala was more than happy to give me time to learn more about what other people. I went to the Air and Trade Show with him and he let me spend an entire afternoon touring the CRT (Compressor Research Test facility) with a friend who works there.

The most important thing I gained from this job was knowledge about computers. I had been computer literate before this, but through working with several different machines and languages, I think that now I've become truly computer competent. I wasn't involved with any research because Mr. Cala wasn't doing at the time. But just by being there, I picked up on techniques and ideas that will be useful in school this fall.

```
0     ! PROGRAM "MULTICOPU" - COPY/PURGE MULTIPLE FILES
20    ! WRITTEN BY GREG CALA
30        ! MODIFIED BY CHRIS HATCH AUG 1990
40              !KOMPRESS, KEEP DATE, AND SHOW HOLES OPTIONS ADDED
50    OPTION BASE 1                 !STARTS SUBSCRIPTS AT 1
60    DIM File_name$(150)[80],Timedate$(150)[200],Date1$[80],Time1$[80]
70    PRINTER IS CRT
80 Prompt:    !
90    PRINT "Do you wish to Copy, Purge, Listcat, Show Holes, Kompress, or Quit
?"
100   INPUT "(C/P/L/S/K/Q)",Ans$
110   IF UPC$(Ans$)="Q" THEN STOP
120   IF UPC$(Ans$)="P" THEN Purger
130   IF UPC$(Ans$)="L" THEN Listcat
140   IF UPC$(Ans$)="C" THEN Copier
150   IF UPC$(Ans$)="K" THEN Kompress
160   IF UPC$(Ans$)="S" THEN Show_holes
170   GOTO Prompt
180 Listcat:   !
190   OUTPUT KBD;":,700,1";
200   LINPUT "Enter msus to CATalog ?",Cat_msus$
210   CALL Numbrd_cat(Cat_msus$,CRT,File_name$(*))
220   GOTO Prompt
230 Copier:!
240   OUTPUT KBD;":,700,0,0";
250   LINPUT "SOURCE msus to COPY from ?",Source_msus$
260   OUTPUT KBD;":,700,1";
270   LINPUT "DESTINATION msus to COPY to ?",Dest_msus$
280   CALL Numbrd_cat(Source_msus$,CRT,File_name$(*))
290 Startcopy:INPUT "Enter FIRST file NUMBER to be COPIED, or 999 to QUIT",F1
300   IF F1=999 THEN Endcopy
310   INPUT "Enter LAST file NUMBER to be COPIED, (may be SAME as FIRST for ONLY
 1 COPY)",F2
320   INPUT "Do you want to keep the old time and date on these files? (Y/N)",An
s$
330   IF UPC$(Ans$)="Y" THEN CAT Source_msus$ TO Timedate$(*)
340   CALL Msus_presuf(Dest_msus$,Dest_pre$,Dest_suf$)
350   CALL Msus_presuf(Source_msus$,Source_pre$,Source_suf$)
360   FOR F=F1 TO F2
370     IF UPC$(Ans$)="Y" THEN GOSUB Date_set
380     COPY Source_pre$&File_name$(F)&Source_suf$ TO Dest_pre$&File_name$(F)&De
st_suf$
390     PRINT File_name$(F);" COPIED  TO ";Dest_pre$&File_name$(F)&Dest_suf$
400     IF UPC$(Ans$)="Y" THEN GOSUB Date_reset
410   NEXT F
420   GOTO Startcopy
430 Endcopy:          !
440   GOTO Prompt
450 Purger:                   !
460   OUTPUT KBD;":,1500,0";
470   LINPUT "msus to PURGE from ?",Purge_msus$
480   CALL Msus_presuf(Purge_msus$,Purge_pre$,Purge_suf$)
490 Startpurge: !
500   CALL Numbrd_cat(Purge_msus$,CRT,File_name$(*))
510   INPUT "Enter FIRST file NUMBER to be PURGED, or 999 to QUIT",F1
520   IF F1=999 THEN Endpurge
530   INPUT "Enter LAST file NUMBER to be PURGED, (may be SAME as FIRST for ONLY
 1 PURGE)",F2
540   FOR F=F1 TO F2
550     PURGE Purge_pre$&File_name$(F)&Purge_suf$
560   NEXT F
570   GOTO Startpurge
```

```
580 Endpurge: !
590   GOTO Prompt
600   STOP
610 Date_set:   !REMEMBERS REAL TIME AND DATE AND SETS CLOCK TO TME AND DATE OF
                 FILES BEING COPIED
620   Realtime$=TIME$(TIMEDATE)
630   Realdate$=DATE$(TIMEDATE)
640   Ff=F+7
650   Date1$=Timedate$(Ff)[56,64]
660   Time1$=Timedate$(Ff)[67,71]
670   IF Date1$="        " THEN
680       SET TIMEDATE DATE("1 JAN 1911")+TIME("11:11:11")
690       RETURN
700   END IF
710   IF Time1$="        " THEN
720       SET TIMEDATE DATE("1 JAN 1911")+TIME("11:11:11")
730       RETURN
740   END IF
750   IF Date1$[1,1]=" " THEN Date1$="0"&Date1$[2,9]
760   Date1$=Date1$[1,2]&" "&Date1$[4,6]&" 19"&Date1$[8,9]
770   SET TIMEDATE DATE(Date1$)+TIME(Time1$)
780   RETURN
790 Enddate_set:    !
800 Date_reset:  !RESETS INTERNAL CLOCK TO REAL TIME AND DATE
810   SET TIMEDATE DATE(Realdate$)+TIME(Realtime$)
820   RETURN
830 Enddate_
840 Kompress: !DELETES HOLES IN A HARD DISK VOLUME
850   PRINT "You cannot compress a hard disk section with more than 5 meg on it.
"
860   PRINT "Do you want to compress using a blank section of the hard disk (5 m
eg max)"
870   INPUT "a ramdisk (10674, 256 Byte Rec max) or quit? (h/r/q)",Ans$
880   IF UPC$(Ans$)="H" THEN Harddisk
890   IF UPC$(Ans$)="R" THEN Harddisk
900   IF UPC$(Ans$)="Q" THEN Prompt
910 Endkompress:    !
920 Harddisk:       !
930   OUTPUT KBD;":,700,0,0";
940   LINPUT "Msus to compress?",Source_msus$
950   CALL Numbrd_cat(Source_msus$,CRT,File_name$(*))
960   IF UPC$(Ans$)="R" THEN GOTO Make_ramdisk
970   OUTPUT KBD;":,700,0,3";
980   LINPUT "Blank msus to use?",Dest_msus$
990 Ram:    !
1000  INPUT "Enter start of range for compression or 999 to quit.",F1
1010  IF F1=999 THEN Prompt
1020  INPUT "Enter end of range for compression.",F2
1030  INPUT "Keep old dates in this section? (Y/N)",Z$
1040  IF UPC$(Z$)="Y" THEN CAT Source_msus$ TO Timedate$(*)
1050  CALL Msus_presuf(Dest_msus$,Dest_pre$,Dest_suf$)
1060  CALL Msus_presuf(Source_msus$,Source_pre$,Source_suf$)
1070  FOR F=F1 TO F2
1080    IF UPC$(Z$)="Y" THEN GOSUB Date_set
1090    COPY Source_pre$&File_name$(F)&Source_suf$ TO Dest_pre$&File_name$(F)&De
st_suf$
1100    IF UPC$(Z$)="Y" THEN GOSUB Date_reset
1110  NEXT F
1120  PRINT "Copy complete"
1130  INPUT "Continue with compression?  Next step purges old files.  (Cont./Qui
t)",Aa$
1140  IF UPC$(Aa$)="Q" THEN Prompt
1150  FOR F=F1 TO F2
1160    PURGE Source_pre$&File_name$(F)&Source_suf$
1170  NEXT F
1180  FOR F=F1 TO F2
```

```
1190     IF UPC$(Z$)="Y" THEN GOSUB Date_set
1200     COPY Dest_pre$&File_name$(F)&Dest_suf$ TO Source_pre$&File_name$(F)&Sour
ce_suf$
1210     IF UPC$(Z$)="Y" THEN GOSUB Date_reset
1220     IF UPC$(Ans$)="P" THEN GOTO Ram1
1230     PURGE Dest_pre$&File_name$(F)&Dest_suf$
1240 Ram1:
1250   NEXT F
1260   PRINT "Compress complete"
1270 Endhercdisk:
1280   GOTO Prompt
1290 Make_ramdisk: !
1300   INITIALIZE ":MEMORY,0",10782      !SETS UP A RAMDISK
1310   Dest_msus$=":MEMORY,0"
1320   GOTO Ram
1330 Endmake_ramdisk: !
1340 Show_holes:                    !
1350   OUTPUT KBD;":,700,0,0";
1360   LINPUT "Enter musu to evaluate.",Cat_msus$
1370   LINPUT "Route to Printer or to Crt?  (P/C)",Dis$
1380   Dis=1
1390   IF UPC$(Dis$)="P" THEN Dis=701
1400   CALL Show_hole(Cat_msus$,Dis)
1410 Endshow_holes: !
1420   GOTO Prompt
1430   END
1440   SUB Numbrd_cat(Msus$,Prt_dev,Name$(*))   !INSERTS NUMBERS IN THE CATALOG
                    AND READS CATALOG INTO AN ARRAY
1450     PRINTER IS Prt_dev
1460     CAT Msus$ TO Name$(*)
1470     Name$(5)="FILE "&Name$(5)[1,71]
1480     Name$(6)="NUM  "&Name$(6)[1,71]
1490     Name$(7)="==== "&Name$(7)[1,71]
1500     FOR F=1 TO 7
1510       PRINT Name$(F)
1520     NEXT F
1530     FOR F=1 TO 9
1540       N=F+7
1550       IF Name$(N)="" THEN Endprt
1560       PRINT "   "&VAL$(F)&" "&Name$(N)[1,71]
1570       Name$(F)=Name$(N)[1,20]
1580     NEXT F
1590     FOR F=10 TO 99
1600       N=F+7
1610       IF Name$(N)="" THEN Endprt
1620       PRINT "  "&VAL$(F)&" "&Name$(N)[1,71]
1630       Name$(F)=Name$(N)[1,20]
1640     NEXT F
1650     FOR F=100 TO 150
1660       N=F+7
1670       IF Name$(N)="" THEN Endprt
1680       PRINT " "&VAL$(F)&" "&Name$(N)[1,71]
1690       Name$(F)=Name$(N)[1,20]
1700     NEXT F
1710 Endprt:  !
1720   SUBEND
1730   SUB Msus_presuf(Msvs$,Msvs_pre$,Msvs_suf$)
1740     OPTION BASE 1
1750     P=POS(Msvs$,":")
1760     IF P<>1 THEN
1770       Msvs_pre$=Msvs$[1,P-1]
1780       IF ((Msvs$[P-1;1]<>"/") AND (Msvs$[P-1;1]<>"\")) THEN Msvs_pre$=Msvs_p
```

```
res&"/"
1790    END IF
1800    Msvs_suf$=Msvs$(F)
1810  SUBEND
1820  SUB Show_hole(Cat$,D)
1830    DIM X$(150)[700],Add$(150)[20],Size$(150)[20],Rec$(150)[20],Add(150),Siz
(150),Rec(150),Gap(152)
1840    CREATE BDAT "CATA:,700,0,3",150,640
1850    PRINTER IS "CATA:,700,0,3";WIDTH 80
1860    CAT Cat$                  !READS THE CATALOG TO THE FILE CATA:,700,0,3
1870    PRINTER IS CRT
1880    ASSIGN @Path1 TO "CATA:,700,0,3";FORMAT ON
1890    ON END @Path1 GOTO Enough
1900    FOR Y=1 TO 150
1910      ENTER @Path1;X$(Y)
1920      IF Y>5 THEN Add$(Y)=X$(Y)[43,47]  !READS ADDRESS OF FILES
1930      IF Y>5 THEN Size$(Y)=X$(Y)[24,28]  !READS REC/FILE OF FILES
1940      IF Y>5 THEN Rec$(Y)=X$(Y)[34,37]  !READS BYTE/REC OF FILES
1950      File=File+1           !COUNTS NUMBER OF FILES IN VOLUME
1960    NEXT Y
1970 Enough:
1980    ASSIGN @Path1 TO *
1990    PURGE "CATA:,700,0,3"
2000    FOR Y=6 TO File
2010      Add(Y)=VAL(Add$(Y))       !CONVERTS ASCII DATA TO NUMERIC
2020      Siz(Y)=VAL(Size$(Y))
2030      Rec(Y)=VAL(Rec$(Y))
2040    NEXT Y
2050    IF Add(6)<>109 THEN         !CHECKS BEGINING OF CATALOG FOR HOLES
2060      Gap(5)=Add(6)-109
2070      Add(5)=109
2080      Siz(5)=0
2090    END IF
2100    FOR Y=6 TO (File-1)
2110      IF Rec(Y)<>256 THEN   !DETERMINES ADDRESS SIZE OF FILE
2120        Zz=(Siz(Y)*Rec(Y)) MOD 256
2130        Siz(Y)=Siz(Y)*Rec(Y)/256
2140      END IF
2150      IF Zz<>0 AND Zz>=.5 THEN Siz(Y)=INT(Siz(Y))+1  !ROUNDS UP DECIMALS
2160      Siz(Y)=INT(Siz(Y))
2170      Zz=0
2180      IF (Add(Y)+Siz(Y))<>Add(Y+1) THEN Gap(Y)=(Add(Y+1)-(Add(Y)+Siz(Y)))
      !CHECKS FOR HOLES
2190      IF X$(Y)[15,18]="BDAT" THEN Gap(Y-1)=Gap(Y-1)-1
2200    NEXT Y
2210    IF X$(File)[15,18]="BDAT" THEN Gap(File-1)=Gap(File-1)-1
2220    PRINTER IS 0
2230    FOR Y=1 TO File       !MAKE PRINTOUT OF CATALOG
2240      PRINT X$(Y)
2250      IF Y>=5 AND Gap(Y)<>0 THEN PRINT USING "K,5D,K,5D";"              HOLE
    ";Gap(Y);"      256     ";(Add(Y)+Siz(Y))
2260    NEXT Y
2270    IF Rec(File)<>256 THEN     !DETERMINES SIZE OF LAST ENTRY
2280      Zz=(Siz(File)*Rec(File)) MOD 256
2290      Siz(File)=Siz(File)*Rec(File)/256
2300    END IF
2310      IF Zz<>0 AND Zz>=.5 THEN Siz(File)=INT(Siz(File))+1!ROUNDS UP DECIMALS
2320      Siz(File)=INT(Siz(File))
2330      Zz=0
2340      Gap(File+1)=(21456-(Add(File)+Siz(File)))!DETERMINES SPACE LEFT ON DIS
K
2350      PRINT USING "K,5D,K,5D";"              HOLE     ";Gap(File+1);"      2
56    ";(Add(File)+Siz(File))
2360      Th=0
2370      FOR Y=1 TO File+1   !TOTALS EMPTY SPACE ON DISK
2380        Th=Th+Gap(Y)
```

```
2390        NEXT Y
2400        PRINT
2410        PRINT USING "K,5D,K,7D,K";"TOTAL HOLES              ";Th;"  *   256 = ";
(Th*256);    BYTES"
2420        PRINT USING "K,2D.D,K";"MAX RECORDS POSSIBLE   21348       ";(1-((Th*25
6)/5465089))*100;" PERCENT USED"
2430        PRINTER IS CRT
2440  SUBEND
```

Acknowledgements

Thank You to:

Gregory Cala for being a good boss and making this an enjoyable experience

Diamion for helping me with D-Base III

Jose for working with me on Tecplot

Ed and John for making room 21 pleasant and for giving me someone to talk to

Chris for teaching me what little Unix I know

Jerry for understanding when I crashed the Iris and for telling how to fix my car

And everybody else in the lab for putting up with my questions and always being willing to help.

APPRENTICESHIP PROGRAM
WRIGHT PATTERSON AIR FORCE BASE
WRIGHT RESEARCH & DEVELOPMENT CENTER

CHET NIETER
cowritten with Matt Bold
MENTOR: DR. WON SOON CHANG

PROJECT: FLASH PLATE EVAPORATOR

The heat pipe was designed in the 1940's to advance the science of heat transfer. The heat pipe has three primary sections: the evaporator, the adiabatic section, and the condenser section. A working fluid in the evaporator is vaporized by incoming thermal energy. The vapor pressure increases and the vapor is forced through the adiabatic section to the condenser where the vapor is condensed and thermal energy is released. The condensed fluid is drawn back to the evaporator section through a porous wick structure. This is made possible by a drop in capillary pressure in the evaporator section of the pipe. In essence, the thermal energy was transferred from the evaporator section of the pipe to the condenser.

Normally the heat pipe is in the form of a long cylindrical pipe used to transfer heat over long distances. A new type of pipe was needed to enhance the ability to collect and remove heat over large surfaces, or to increase the radiation area.

The flash plate evaporator is the third unit in a three unit system designed to remove large quantities of thermal energy from tightly packed electronics primarily on board spacecraft. The flat plate heat pipe, and the side flow heat pipe, the first two units were tested last year. The flat plate removes the thermal energy from the electronic units to be cooled and the side flow transfers the energy to the flash plate where it is radiated away from the craft.

The performance of the flash plate was the primary concern of our research this year. Its effects of both forced and free convection were studied. Effects of gravity and the loss of energy

through the insulation and edge effects were also a concern.

The plate's facing was fabricated form aluminum as was the wick and honeycomb structure inside. The working fluid was acetone. The plate has the dimensions of 11 3/8" x 11 5/8" with a thickness of .36".

The first tests run on the flash plate were free convection tests. The flash plate was heated by 11" x 3" foil resistance heater. Temperature readings were taken every five minutes from eleven thermocouples placed on the flash plate to a Fluke 2286A Datalogger. Typical tests involved starting the heater at 30 watts and letting the flash plate reach steady state, then increasing the power another 10 watts and again allowing the plate to reach steady state. A maximum power test was run to see how much power the plate would transfer at its safe operating temperature of 75 degrees C, this was found to be 80 watts. Tests were also run while tilting the plate at various angles to determine the effects of gravity on the plate's operation; tilts of 90 degrees, 45 degrees, 30 degrees, and 15 degrees were used. The temperature readings were then plotted into graphs using Lotus 123. Using these graphs, irregularities were looked for in the data (fig a).

The set up for the forced convection tests were different. A 11 1/2" x 11 1/2" copper cool block was placed on the condenser side of the plate. A refrigeration and circulation unit was used to circulate SR-1 Dowtherm heat exchange fluid. Six thermocouples were attached to the evaporator side of the flash plate, a thermocouple probe was placed to read the temperature of the cool unit, the temperature of the flow into the cool block, and the

temperature of the flow out of the cool block. Also connected to the Fluke was a flowmeter. As with the free convection, the temperature readings were graphed using Lotus (fig b).

The amount of heat lost during the free convection test was calculated using the following equations.

$$Q=hA(\Delta T)$$

The "h" is found using the following equation;

$$h=\frac{Nu_l K}{L}$$

"Nu" is found using;

$$Nu_l=[0.825+\frac{0.387Re_l^{1/6}}{[1+(0.4121\times Pr)^{9/16}]^{8/27}}]^2$$

and "Re" is found using;

$$Re_l=\frac{g\beta(\Delta T)L^3}{V\alpha}$$

It was found that there was a small loss of energy probably due to edge effects and loss back through the insulation.

Two other projects were worked on during the summer. The study of the effects of magnetism on the capillary pressure of liquid oxygen was started. We assisted in the construction of a condenser for the liquid oxygen. Because of the volatility of liquid oxygen it was not feasible to keep large quantities of the substance in the lab.

We also attempted to construct a pipe using unidirectional nickel wick. A square pipe was first constructed and work was begun on a method for creating a round pipe.

Further development of the flash plate and the flat plate system will allow spacecraft to carry more electronics or more advanced electronics that produce larger quantities of thermal energy. Liquid oxygen working fluids could be assisted by a magnetic field increasing the capacity of the pipe that it is used in. The nickel wick can be used in advanced diode heat pipes in the future. More research must still be done but all of the goals are feasible.

BIBLIOGRAPHY

Chi, S.W. Heat Pipe Theory and Practice, New York: Mcgraw-Hill Book Company, 1976, pp.1-94.

Dunn, P.D. and Reay, D.A. Heat Pipes, Oxford: Pergamon press, 1982, pp. 1-10.

Fleischman, G.L. High Power Spacecraft Thermal Management, Hughes Aircraft Company, electron dynamics division, 1987, pp.55-61.

ACKNOWLEDGEMENTS

The work done was sponsored by the Aero Propulsion and Power Laboratory at the Air Force Wright Research and Development Center. I would like to show my appreciation to the following people: Dr. Won Soon Chang, Brian Hager, Mike Ryan, Don Brigner, John Tennant, Don Reinmuller, and Joella Pinckney.

FLASH PLATE EVAP

JUL 17 TEST NO.FL00130

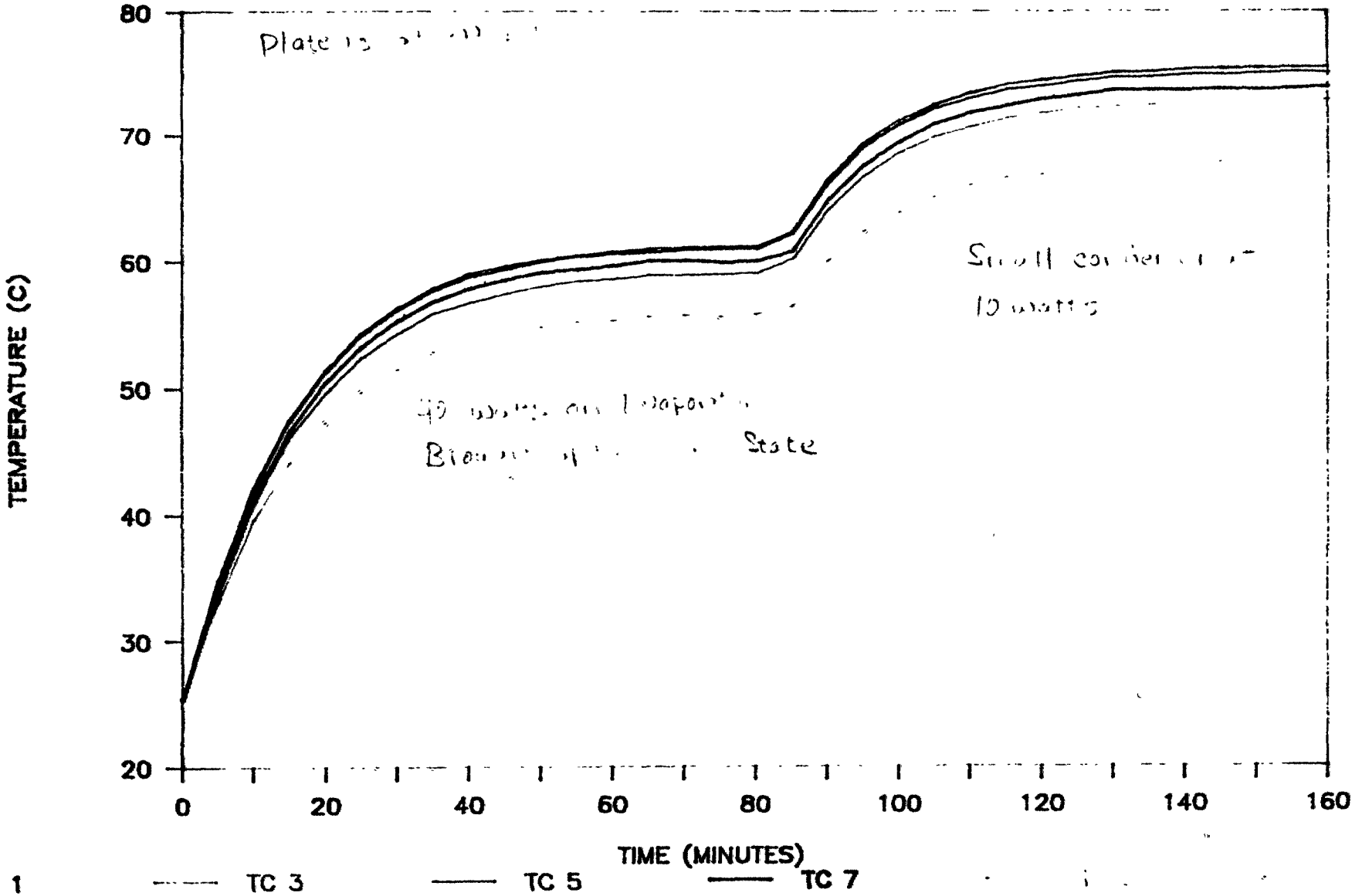


fig a

FLASH PLATE EVAP

AUG 8 TEST NO.FL00180

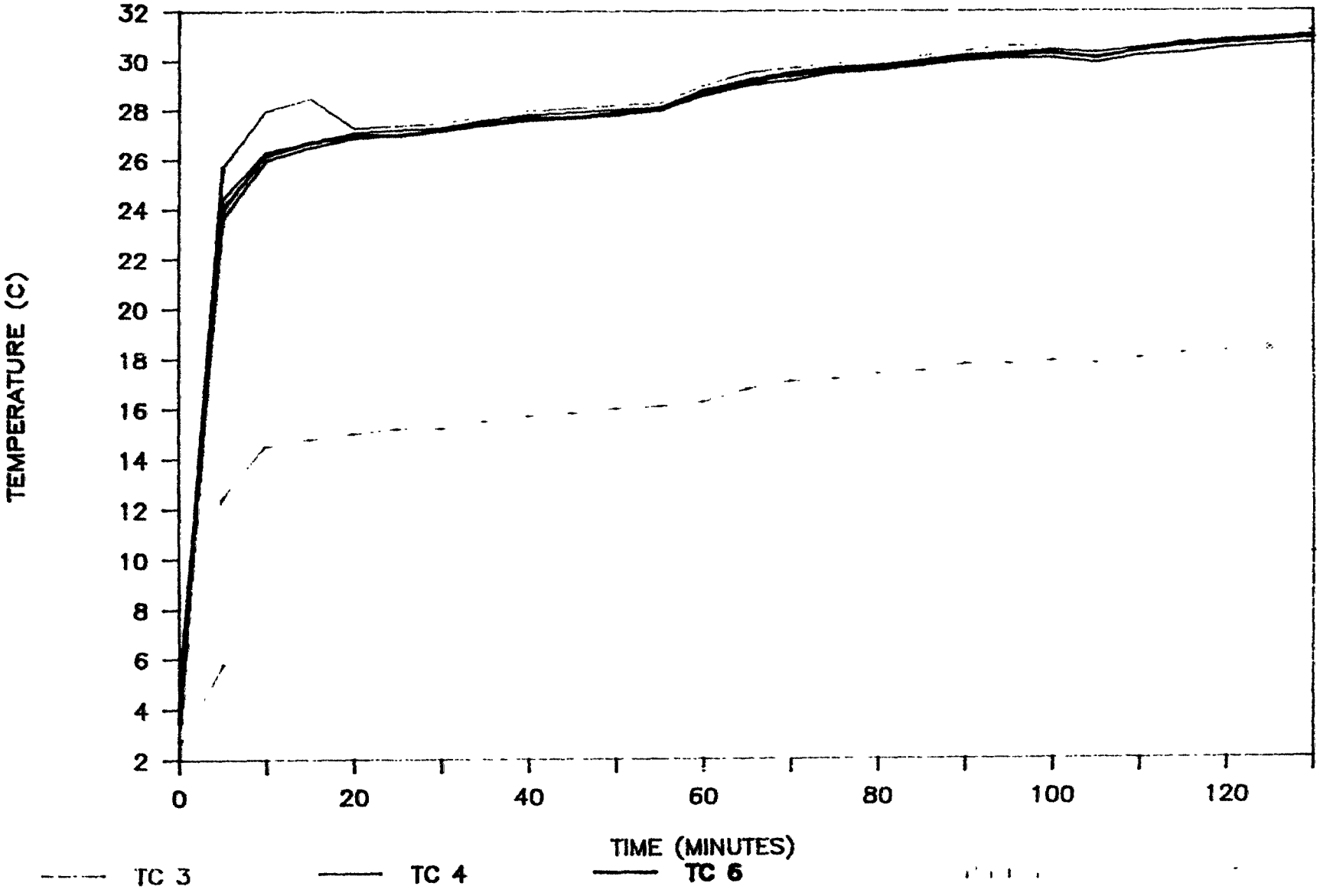


fig b

FINAL REPORT TO UES

Jennifer Pollock
25 August 1990
Em. # 15-9951

ACKNOWLEDGMENTS

I would like to take this opportunity to thank all the people who have taken the time to listen to me, help me, and if they didn't know the answer to my questions, they found someone who did. I couldn't have made it without these people. I would especially like to thank my mentor, Douglas Rabe, as well as: Cindy Aber, Sgt. Scott Allen, Carolyn Cummings, Charles Elrod, David Engler, Edward Fitch, James Flahive, Lori Gilbert, Donald Hoying, Paul Kerney, Capt. Kevin Kilpatrick, Angela Korpi, Janice McCloskey, JoAnne Payne, Pamela Slightham, Richard Taylor, Jess Underwood, Bob Van Hook, Albert Wang, and Christopher Worland. Thanks so much to you all.

FINAL REPORT TO UES

I. Introduction

The following is my final report on the work that I have done while at the Compressor Research Facility, Wright-Patterson AFB, Ohio. I was employed here from 18 June to 17 August 1990. This report includes descriptions of all major tasks that I have completed.

II. A. Computer Systems

The first week of work was when I was taught the majority of the computer systems that I have used. I was taught how to use T-PLOT, on the MAIN system, PV-WAVE, on the VAX system, and ENABLE Spreadsheet/Graphics, on the Zenith Z248 PC. I was required to use the three systems to produce plots of data from compressor tests that were performed at this facility. My first plot was created with the T-PLOT program [1] This program uses data from the Real-Time Data Tapes found at the CRF to produce a plot that is defined by the operator. I then reproduced the same plot[2] using the VAX system so that I could compare one format to the other. This gave me some practical experience on both systems.

B. F119 Programming

After experimenting with the MAIN and VAX systems, I was asked to use the ENABLE Spreadsheet program on the Zenith Z248 PC to produce several plots. To create these plots I had to hand enter over 1,500 figures and equations. The figures I received were from the CRF test of the F119 compressor that was done from 22 March until 6 June 1990. The figures are stored in the Test Mensor Data Books. I then proceeded

to use a program called KERMIT to take the figures from the spreadsheet on the disk and put them onto tape in the VAX system. After that, I was able to write a program that produced plots of specific data.[3] I used these plots to find errors in the raw data in the Mensor Data Books. After correcting several errors I printed the plots again, this time using the corrected data. Over the next week I continued to make changes in the program, and the plots, until they finally were exactly as they were supposed to be.

C. XTE 65 Plots

The next project I was given was to write another program. This program would also be written on the Zenith Z248 PC and was to use the ENABLE Spreadsheet/Graphics program. The data to be plotted was from the XTE 65 test[4], and it would be of certain selected Mensors and Deleted Mensors that are in the Test Mensor Data Books for the XTE 65 test that was run from 10 July to 31 July 1990. I was required to set up the format for these plots and to correct errors in the data. All of this took several days to complete. After completing this program I had to update the data every day and produce plots of the combined data so that trends in the acquisition of the data or in the data itself could be easily seen.

D. Control Room Activities

I had my first chance to do work in the Control Room when engineers Chris Worland and Ed Fitch asked for my assistance in performing a check that is done before all compressor tests begin. This procedure is called a 'channelization check' or an 'end-to-end

check'. This was very interesting and I learned a great deal from it. This check is made to ensure that all sensor lines coming from the compressor to be tested are taking accurate readings. If this check was not performed it would be impossible to consider all the readings as accurate, and that would mean that the test results could be disreputed by the company or its competitors.

E. BASIC program

The final task that I was given was to write my own program, unassisted, in BASIC. This program finds the percent deviation of the pressure ratio, efficiency, and bellmouth and venturi massflows. To write this program, I had to get equations from another program that was in the language of FORTRAN. I then transfered the equations into BASIC. This was especially difficult because I do not know FORTRAN. I worked on this program for the last month of my employment at the CRF. It is now going to be used to help some of the engineers correct potential problems before they can cause serious damage[5].

III. Conclusion

Though my time at the Compressor Research Facility was short, I have learned many things. I hope that I was able to help the people at the CRF with whom I worked, and I am very grateful to them for all of their patience, and assistance, and to you at Universal Energy Systems for the opportunity to work with these people.

REFERENCES

1. See plot 'Compressor Performance'.

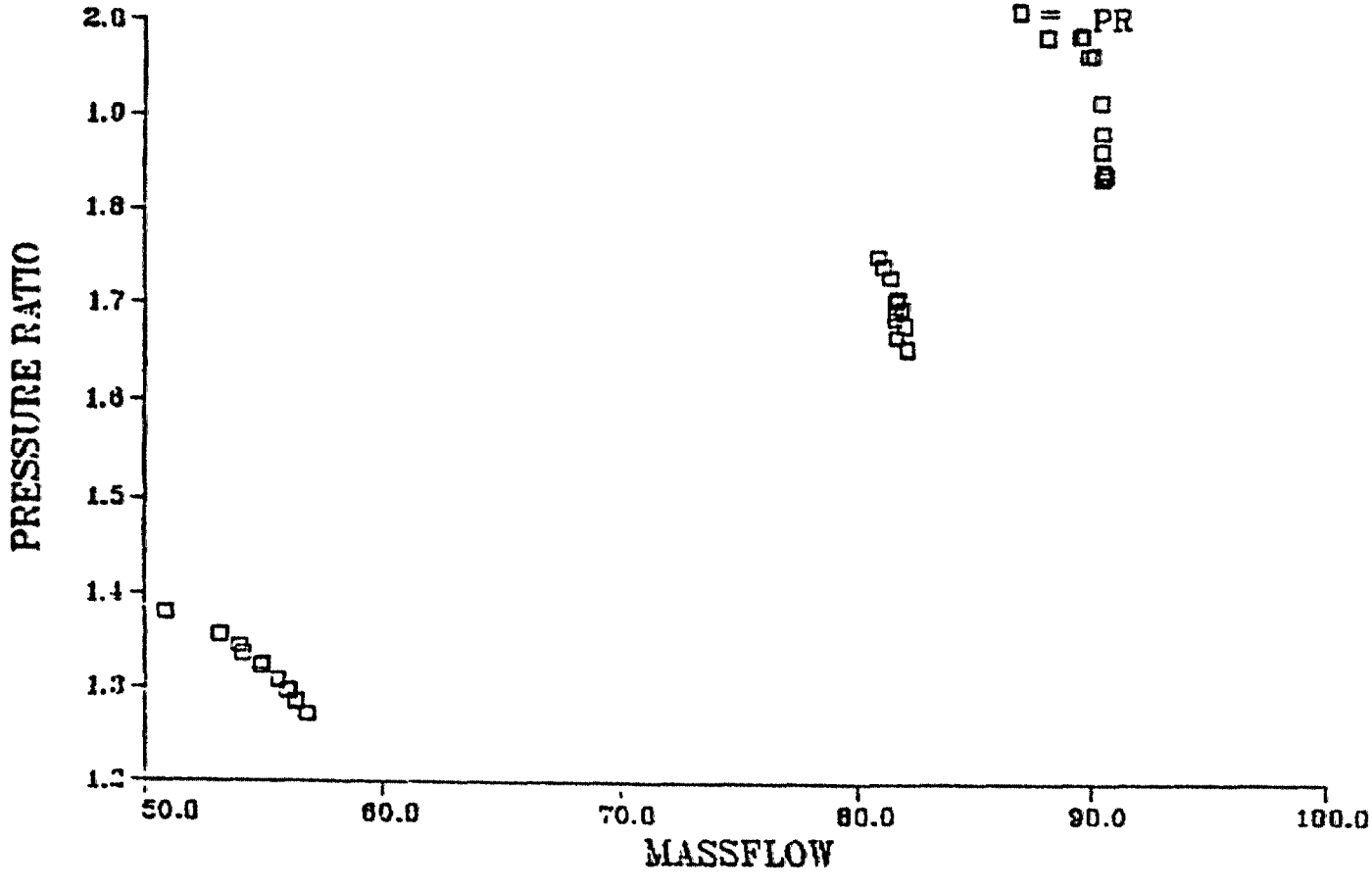
2. See plot 'NAFCOT Compressor Performance'.

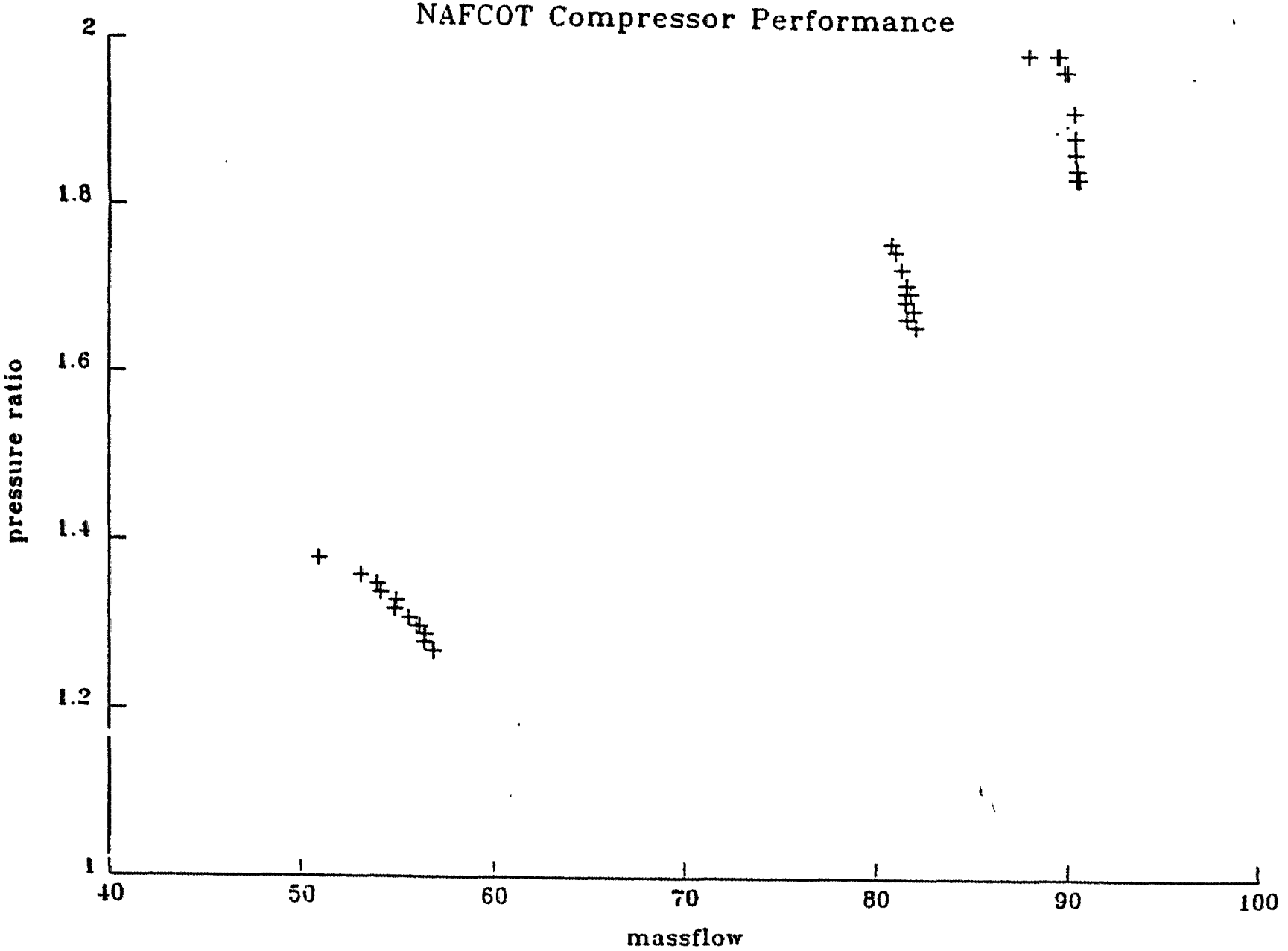
3. See plots labeled 'Group 1'.

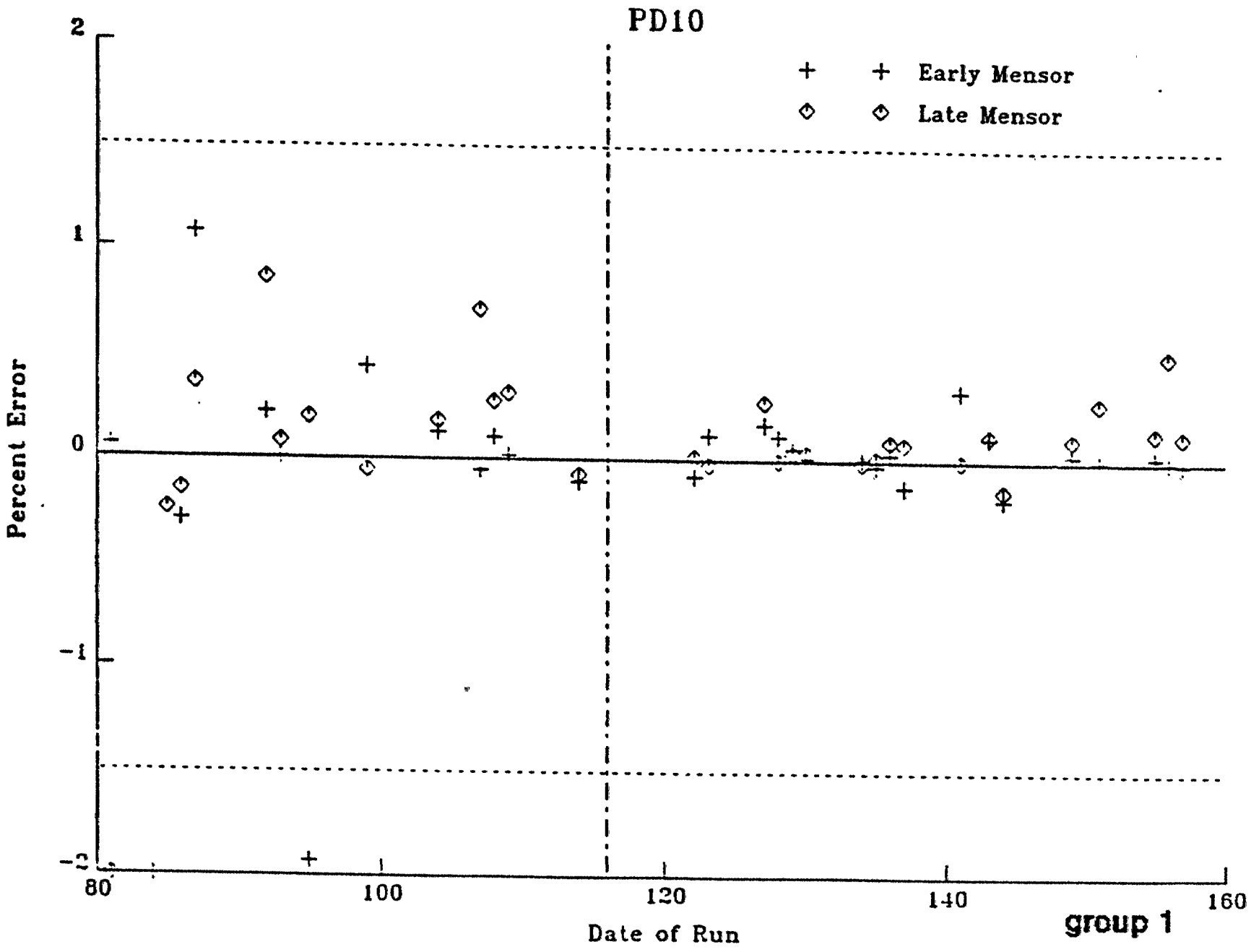
4. See plots labeled 'Group 2'.

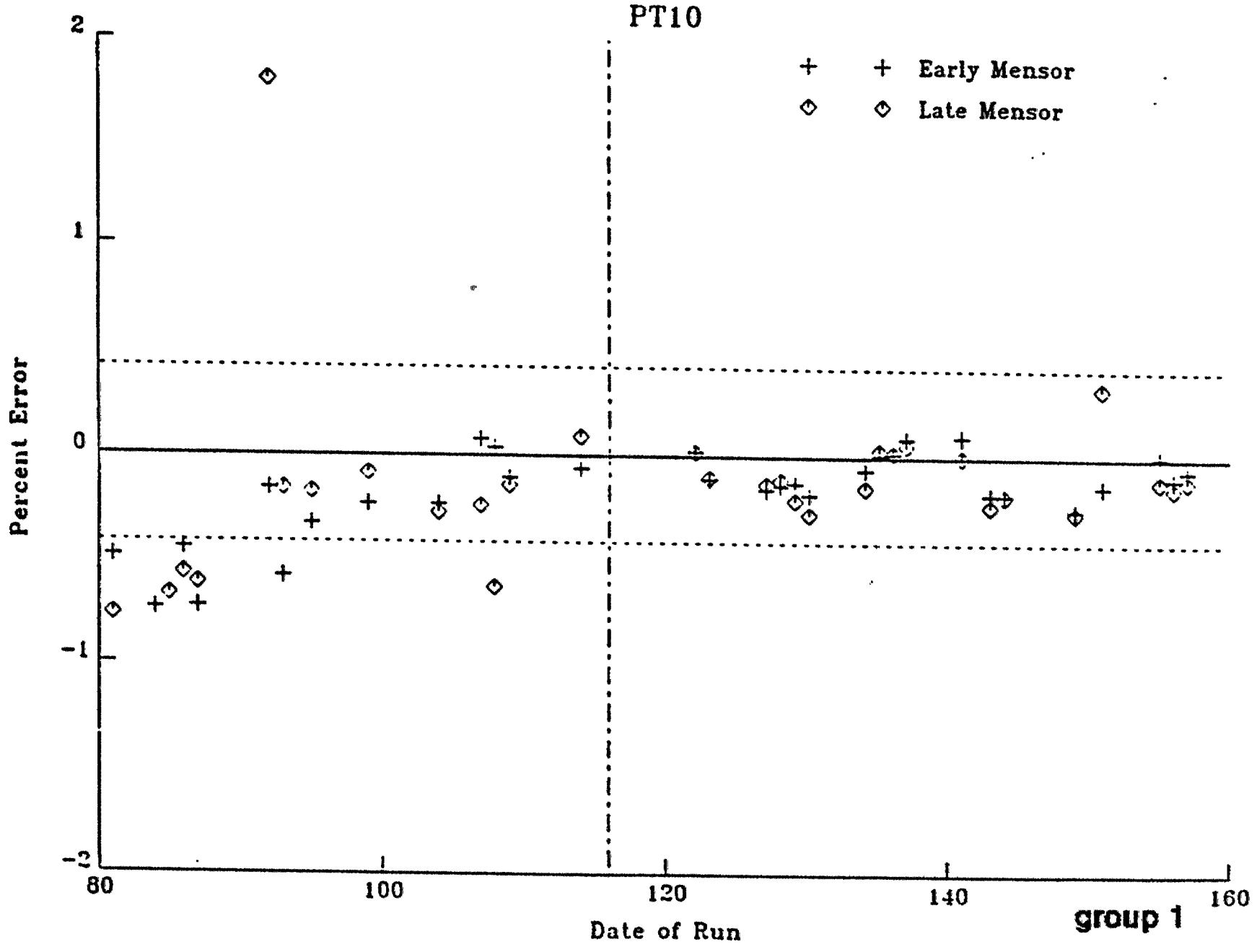
5. See plots labeled 'Group 3'.

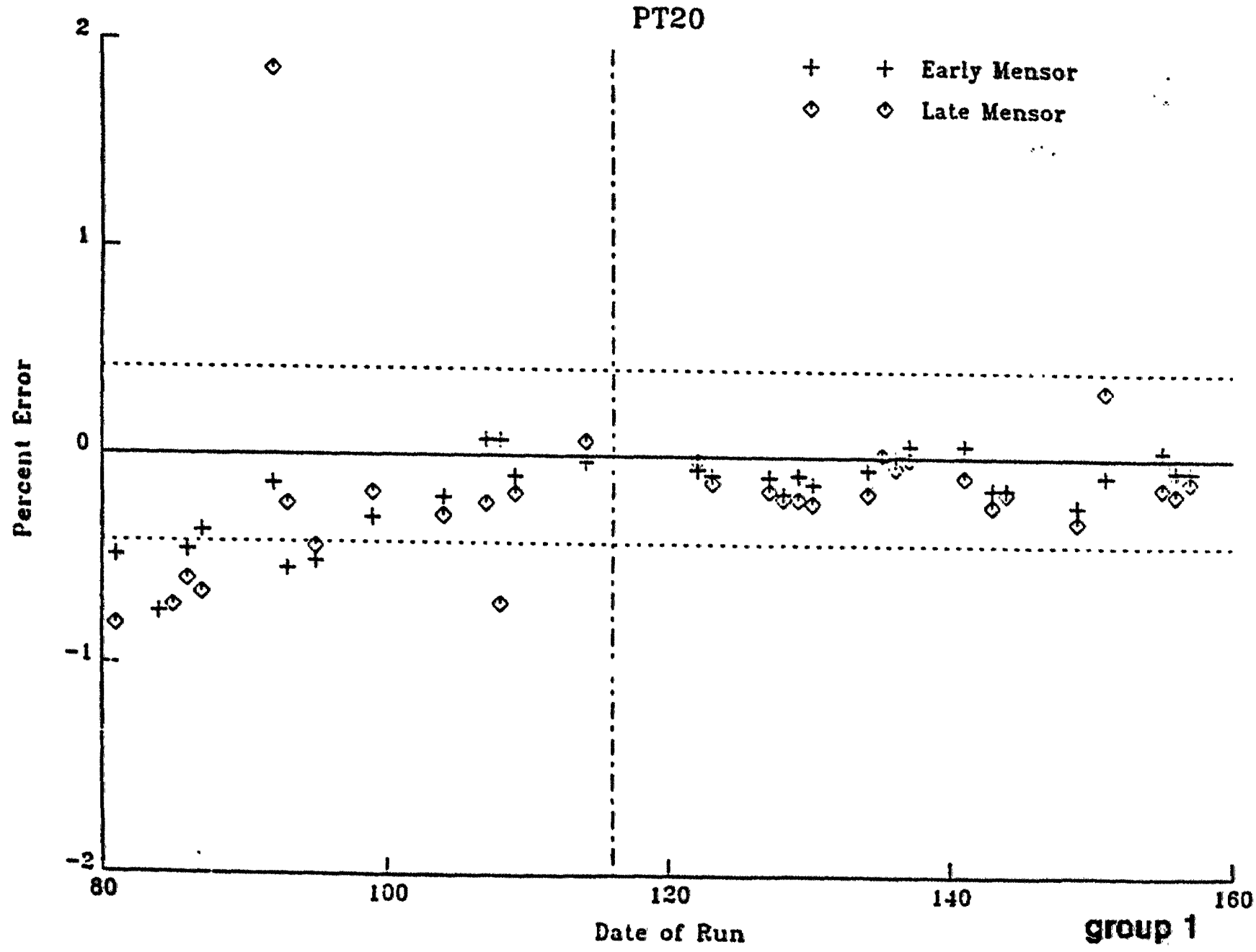
COMPRESSOR PERFORMANCE
NAFCOT AND BASELINE COMPARISON

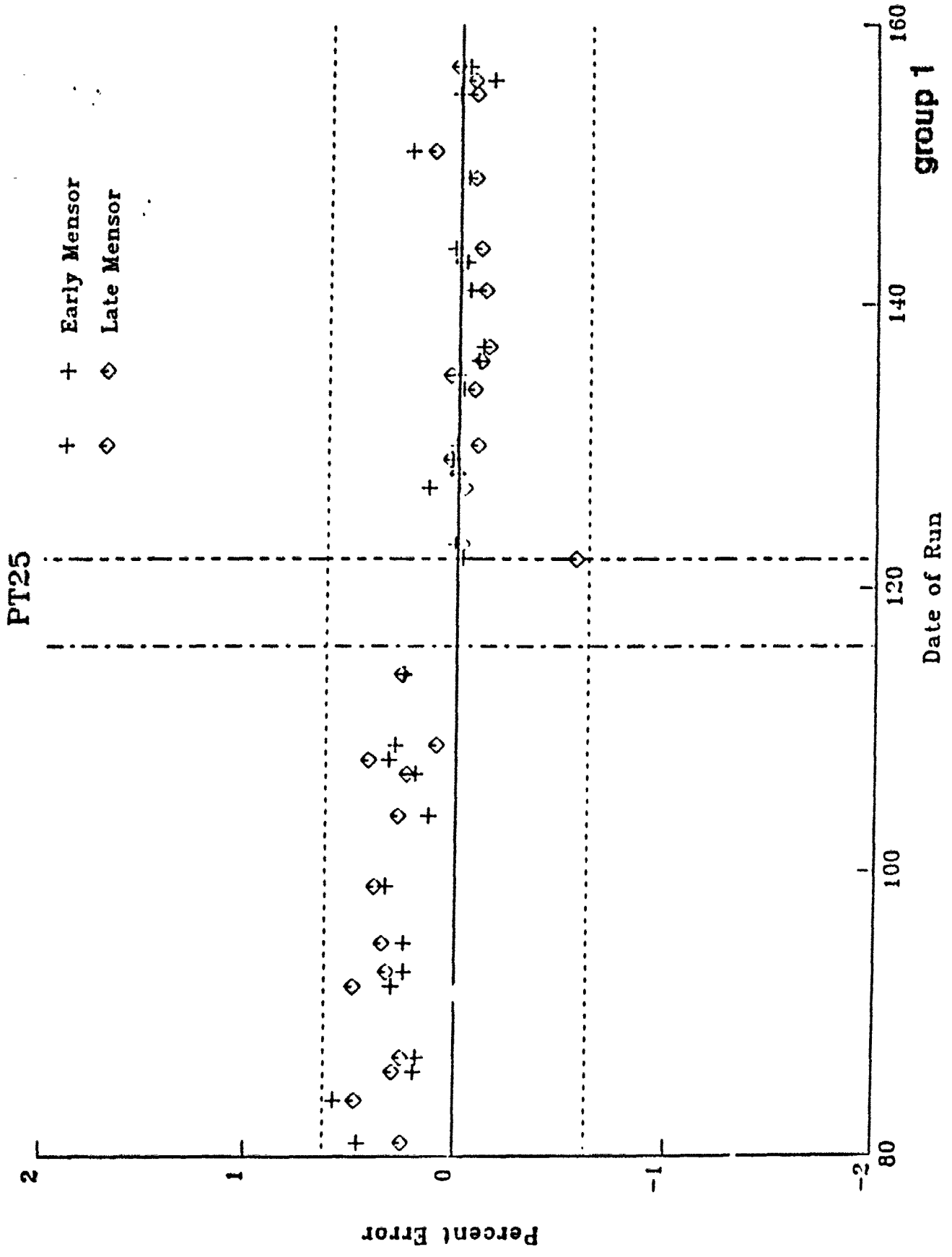


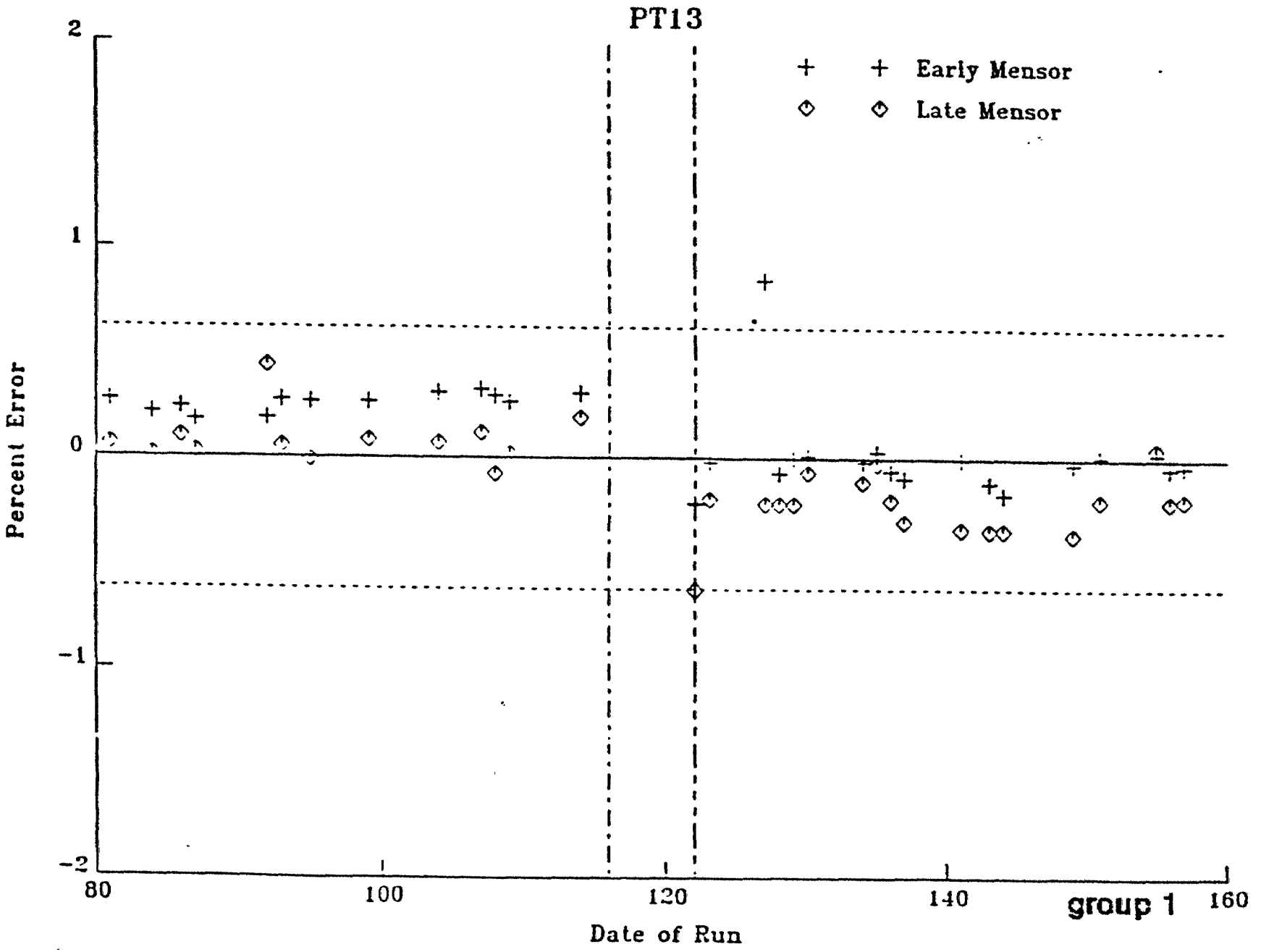

NAFCOT Compressor Performance
pressure ratio
massflow
2
1.8
1.6
1.4
1.2
1
40
50
60
70
80
90
100

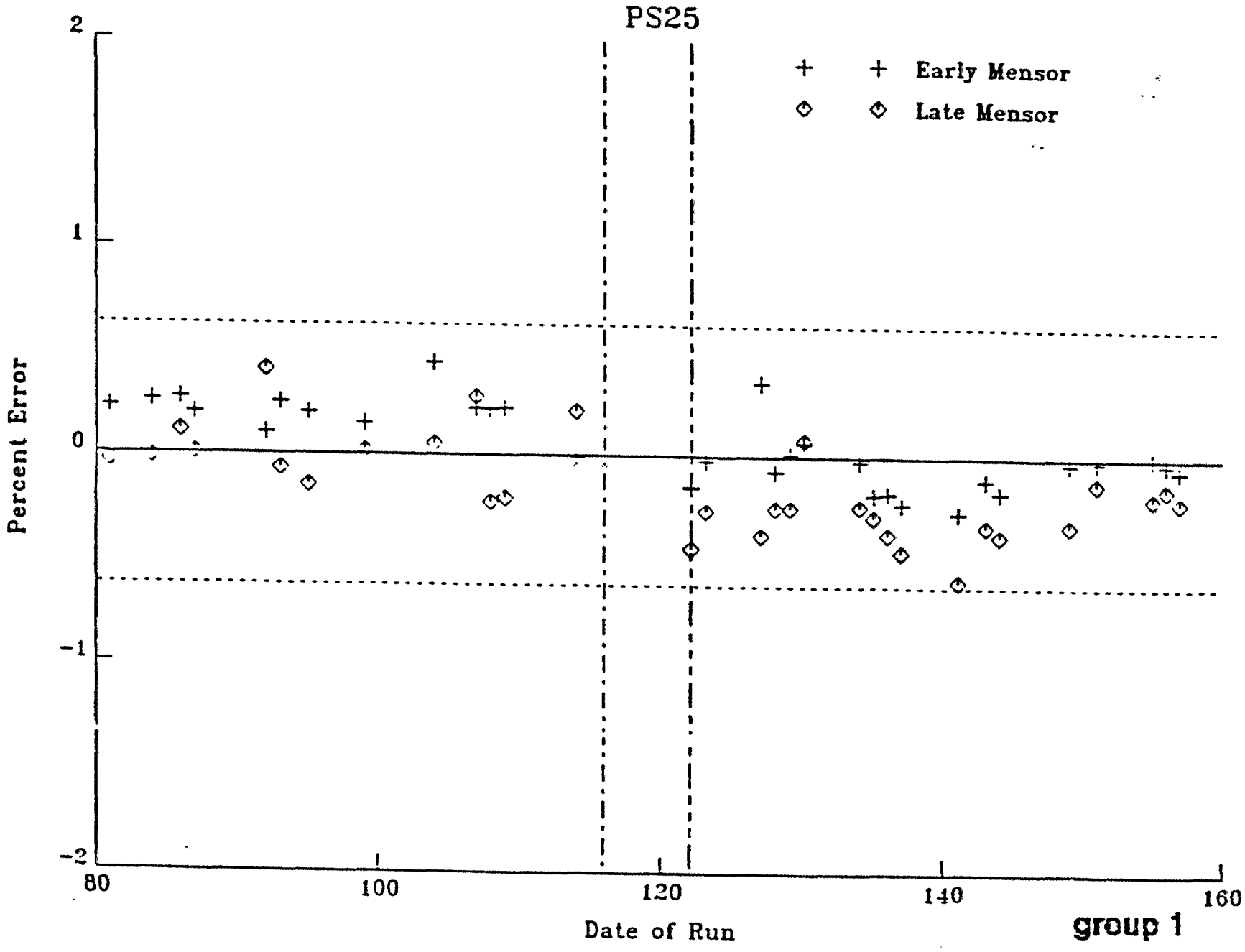

PD10
Early Mensor
Late Mensor
Percent Error
2
1
0
-1
-2
80
100
120
140
160
Date of Run
group 1

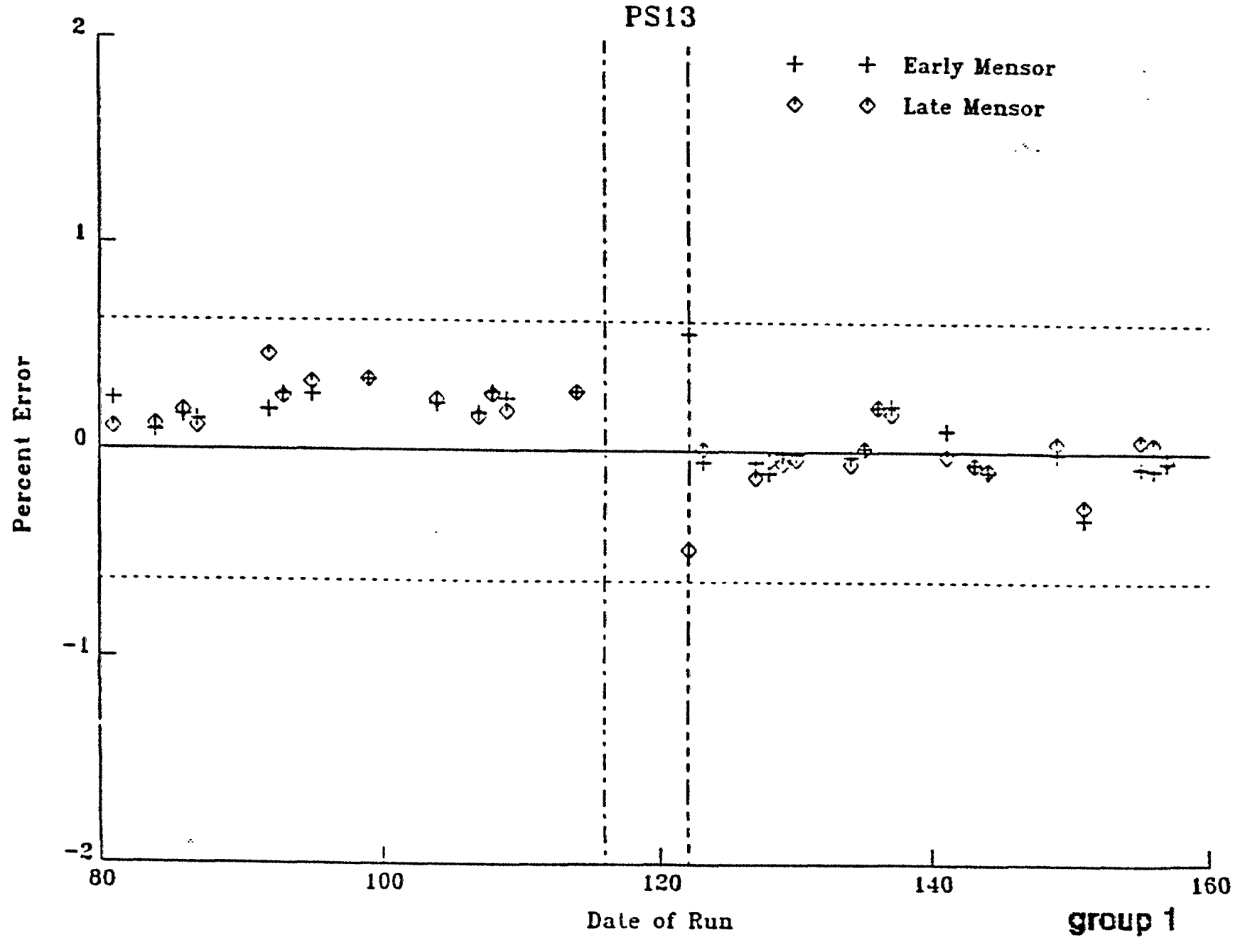

PT10
Early Mensor
Late Mensor
Percent Error
2
1
0
-1
-2
80
100
120
140
160
Date of Run
group 1


PT20
Early Mensor
Late Mensor
Percent Error
2
1
0
-1
-2
80
100
120
140
160
Date of Run
group 1

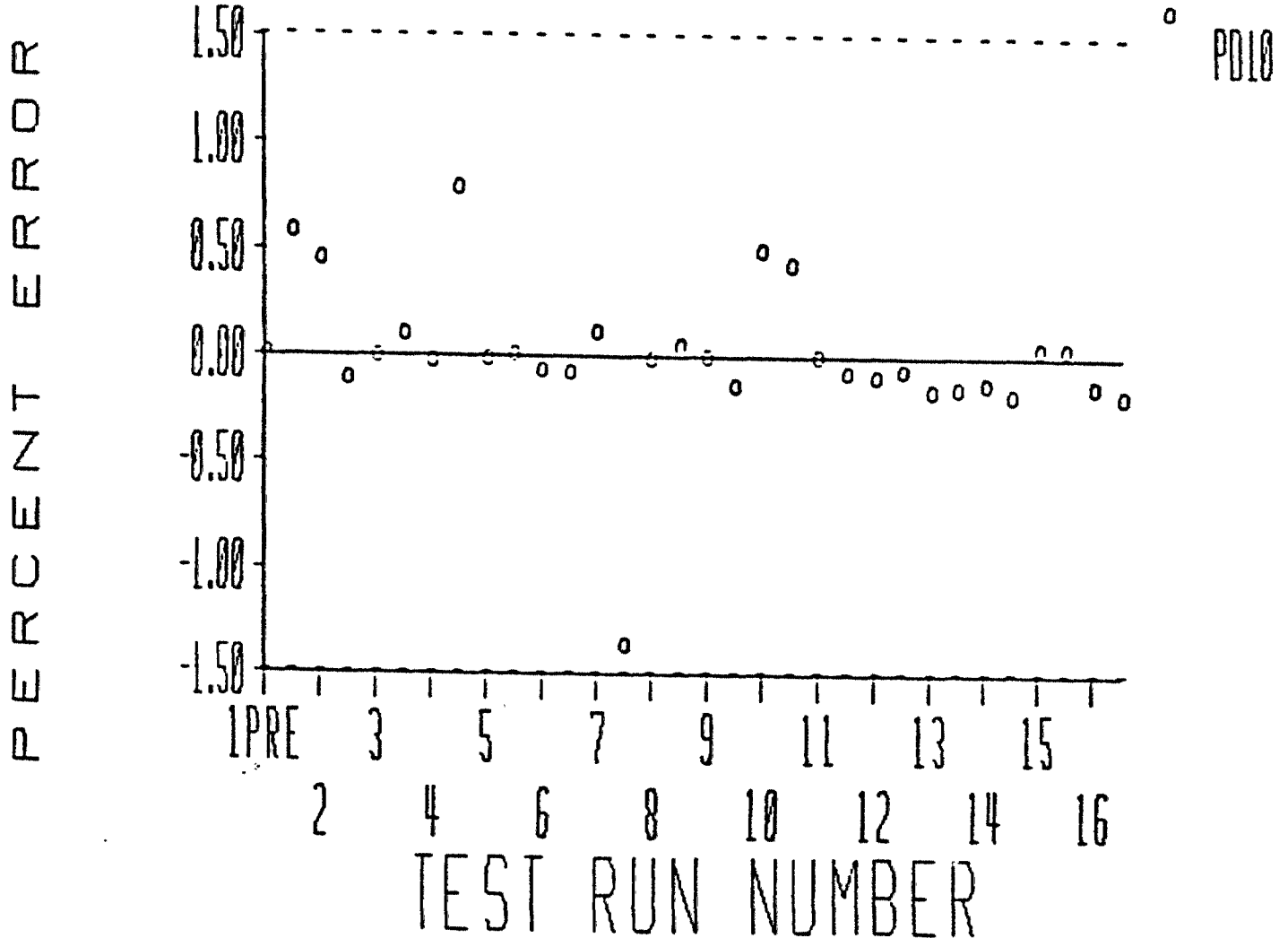

Early Mensor
Late Mensor
PT25
group 1
Percent Error
Date of Run
2
1
0
-1
-2
80
100
120
140
160


PT13
Early Mensor
Late Mensor
Percent Error
2
1
0
-1
-2
80
100
120
140
160
group 1
Date of Run


PS25
Early Mensor
Late Mensor
Percent Error
2
1
0
-1
-2
80
100
120
140
160
Date of Run
group 1

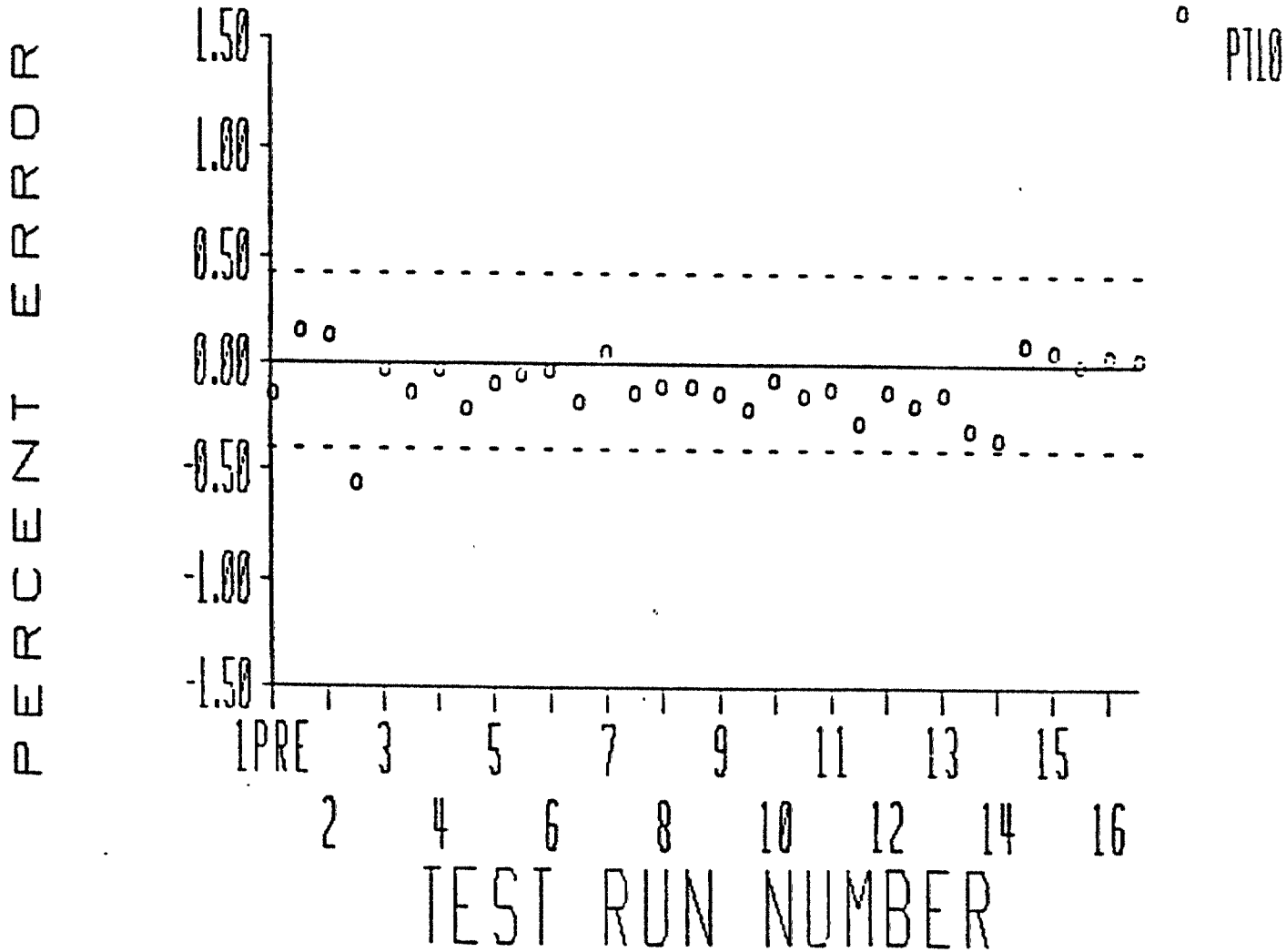

PS13
Early Mensor
Late Mensor
Percent Error
2
1
0
-1
-2
80
100
120
140
160
Date of Run
group 1

TEST MENSOR DEVIATION



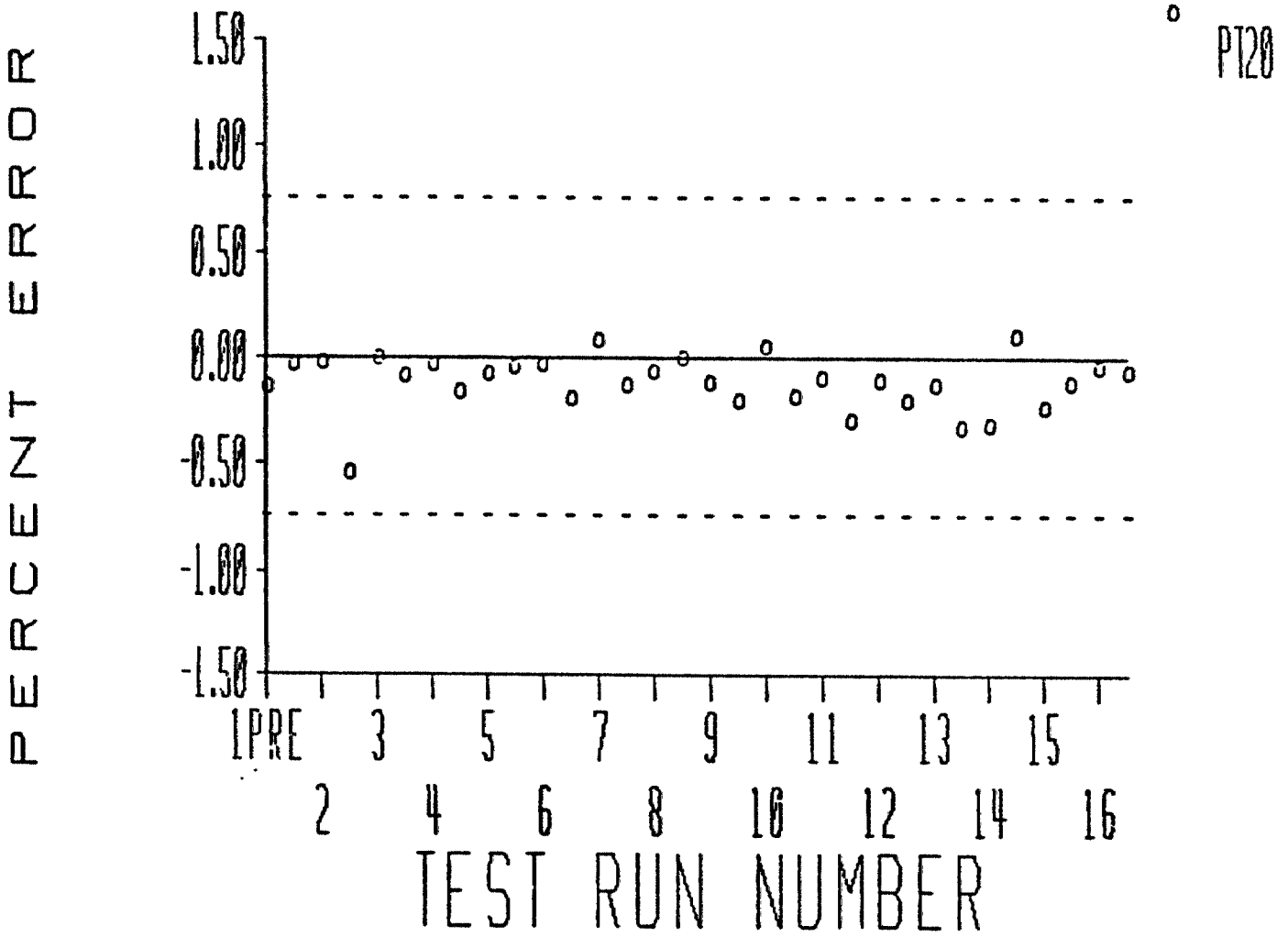
group 2

TEST MENSOR DEVIATION



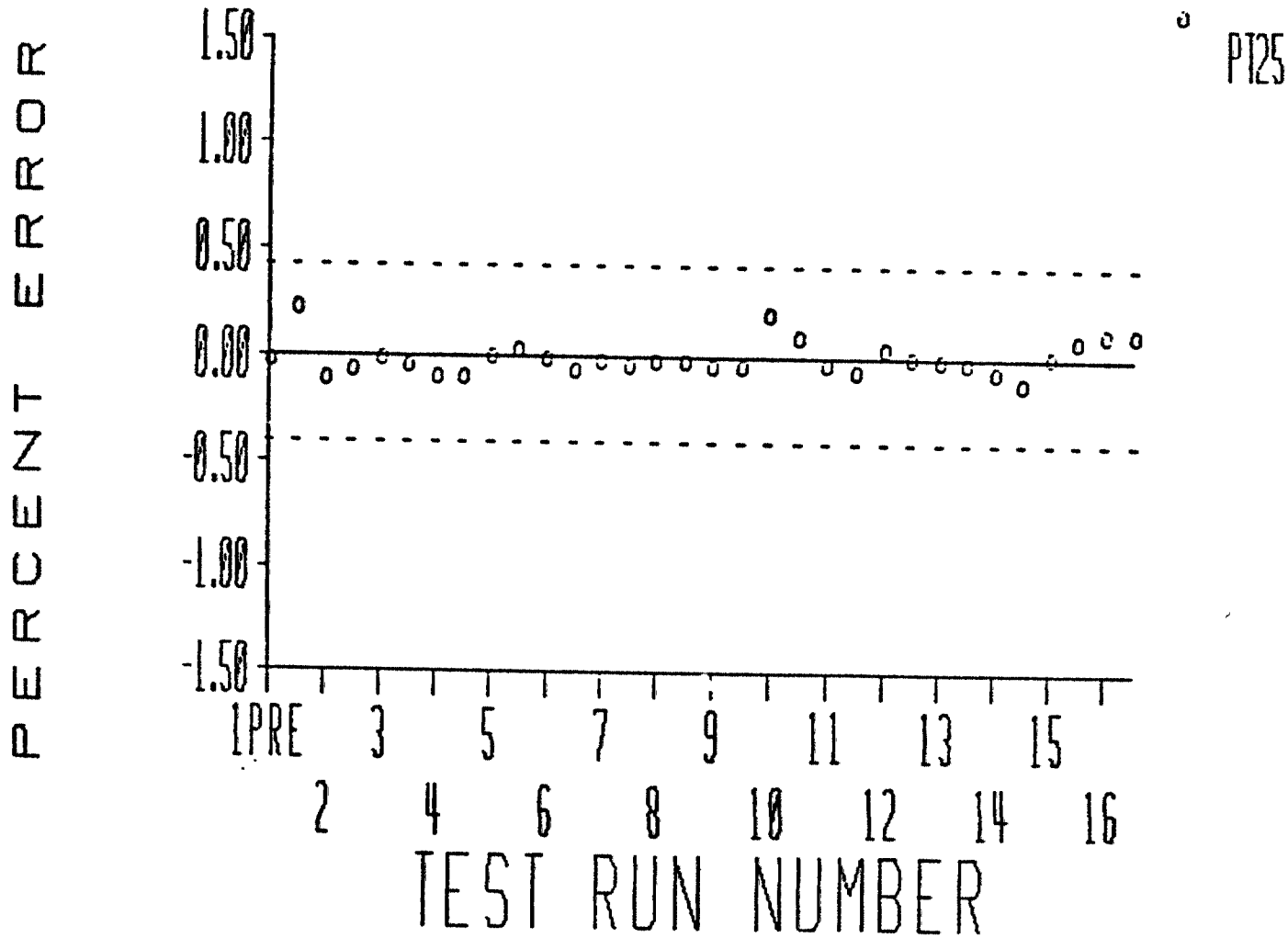
group 2

TEST MENSOR DEVIATION

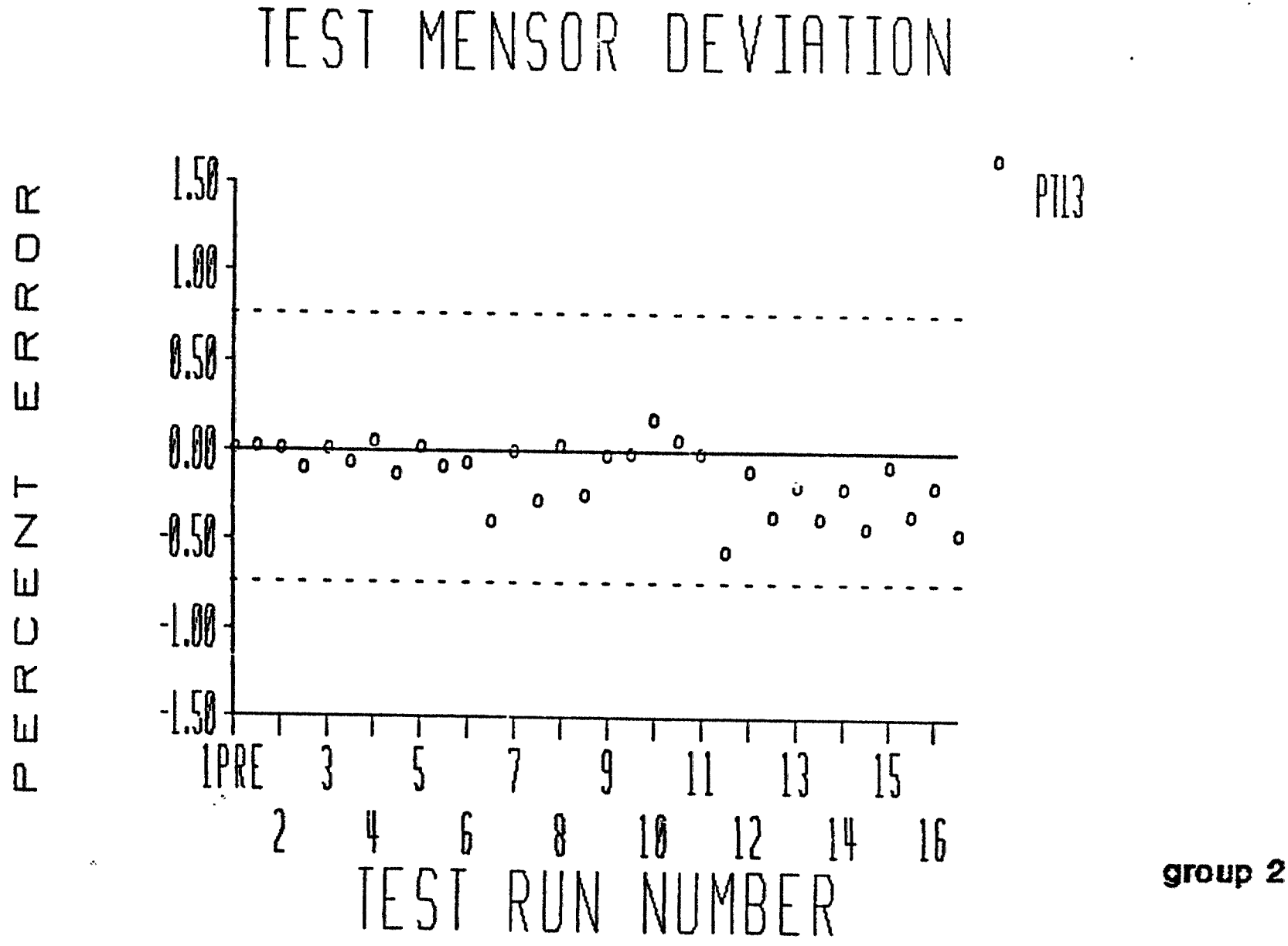


group 2

TEST MENSOR DEVIATION

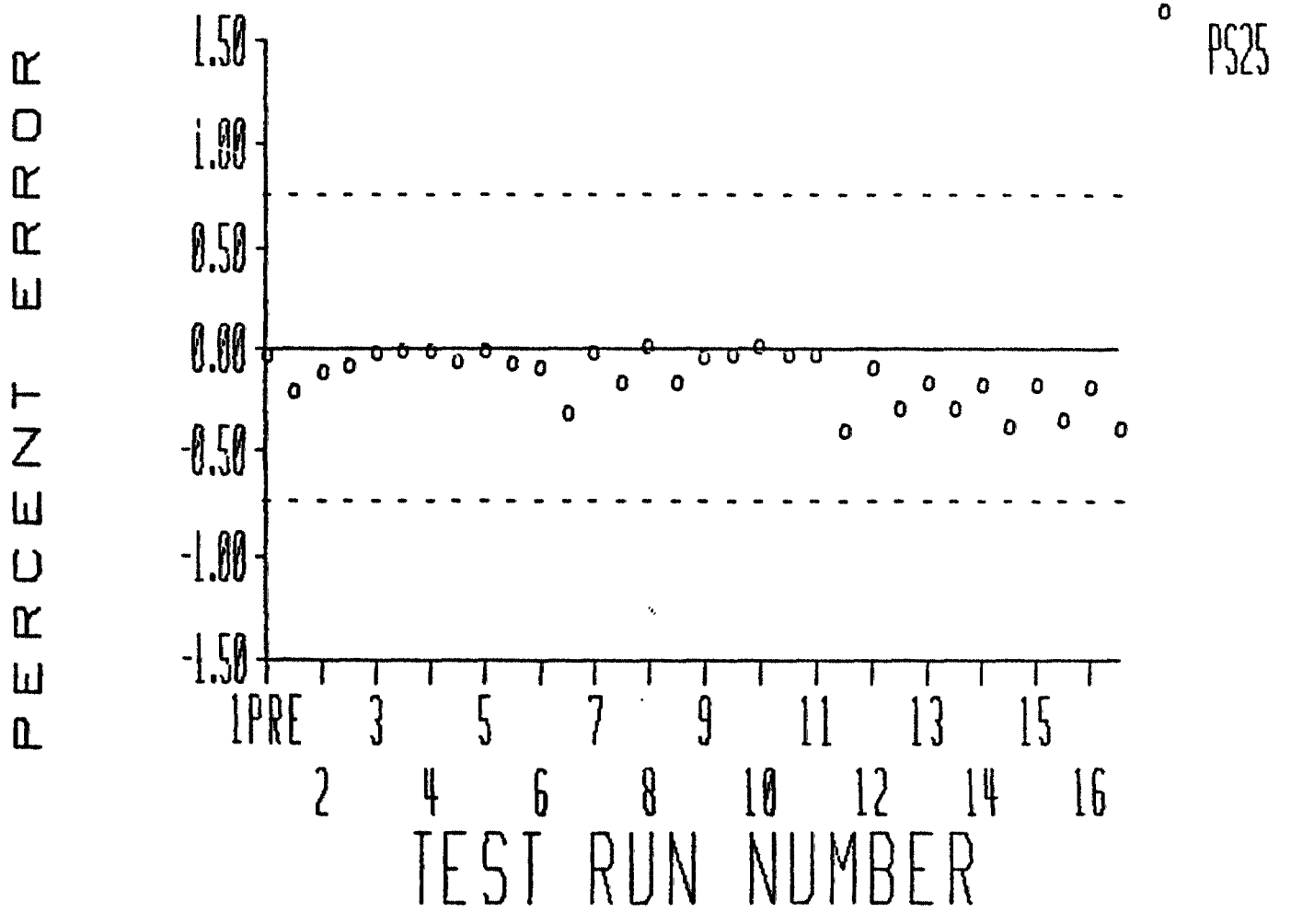


group 2


TEST MENSOR DEVIATION
PERCENT ERROR
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
PT13
1PRE 3 5 7 9 11 13 15
2 4 6 8 10 12 14 16
TEST RUN NUMBER

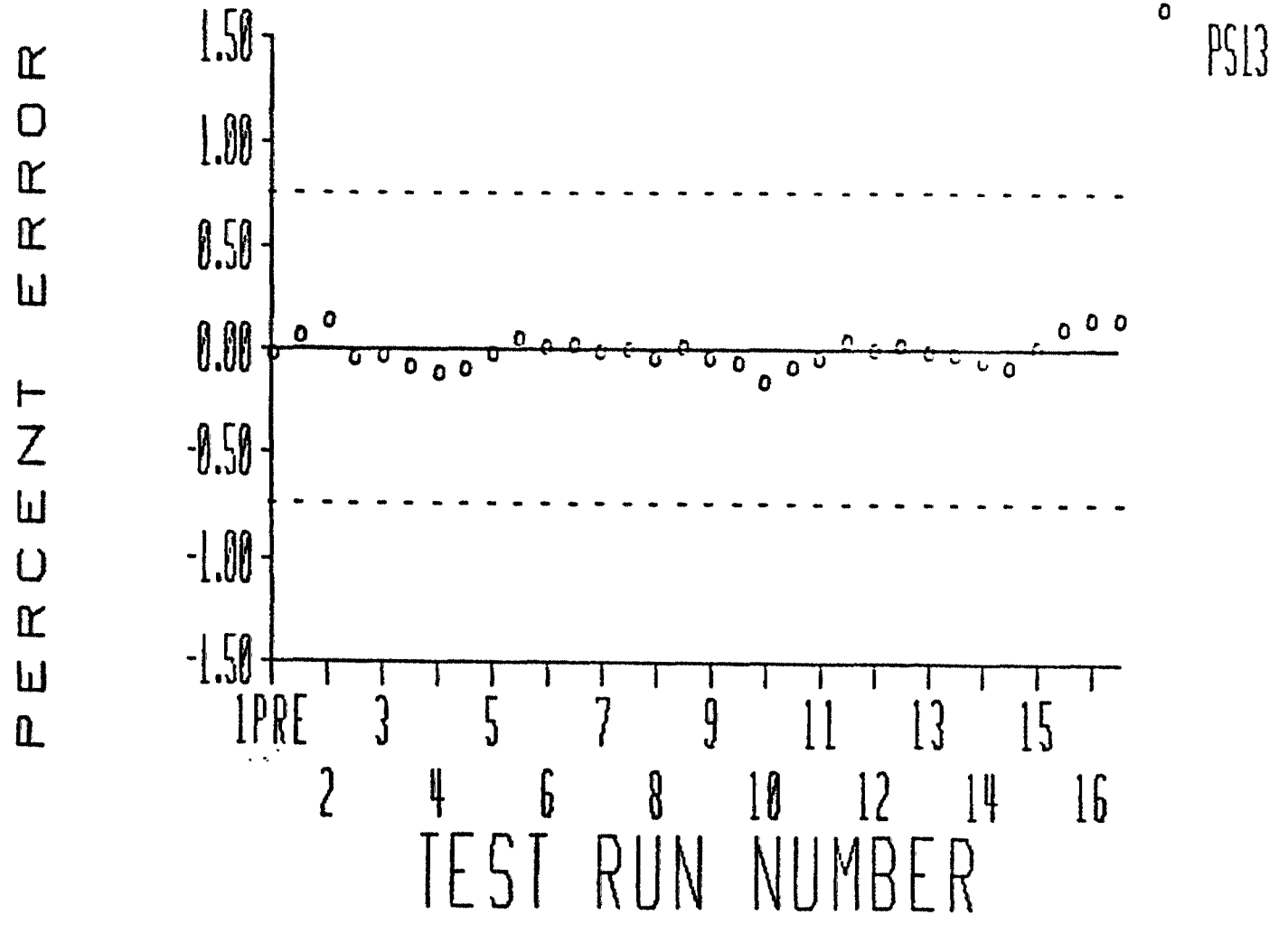
group 2

TEST MENSOR DEVIATION

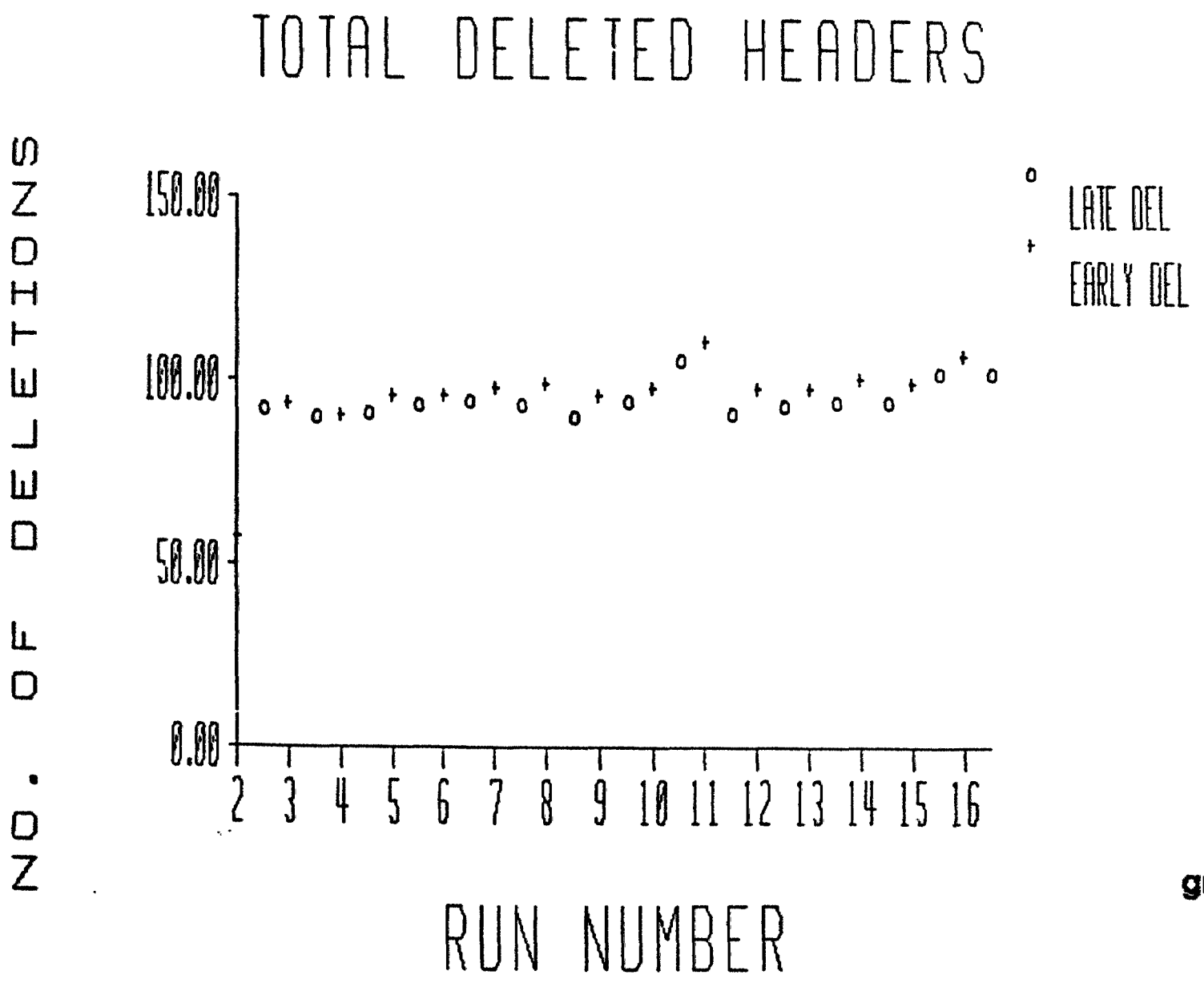


group 2

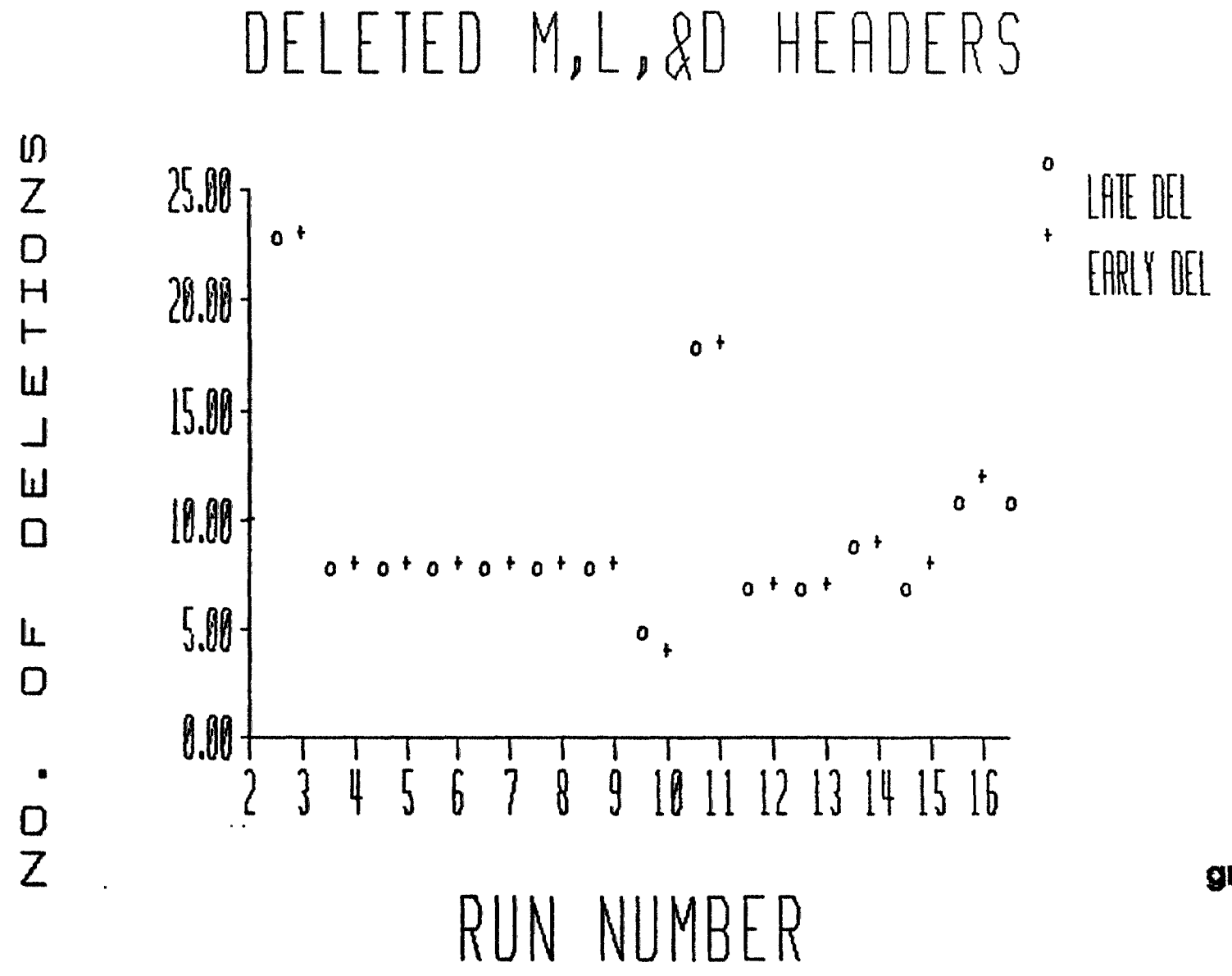
TEST MENSOR DEVIATION



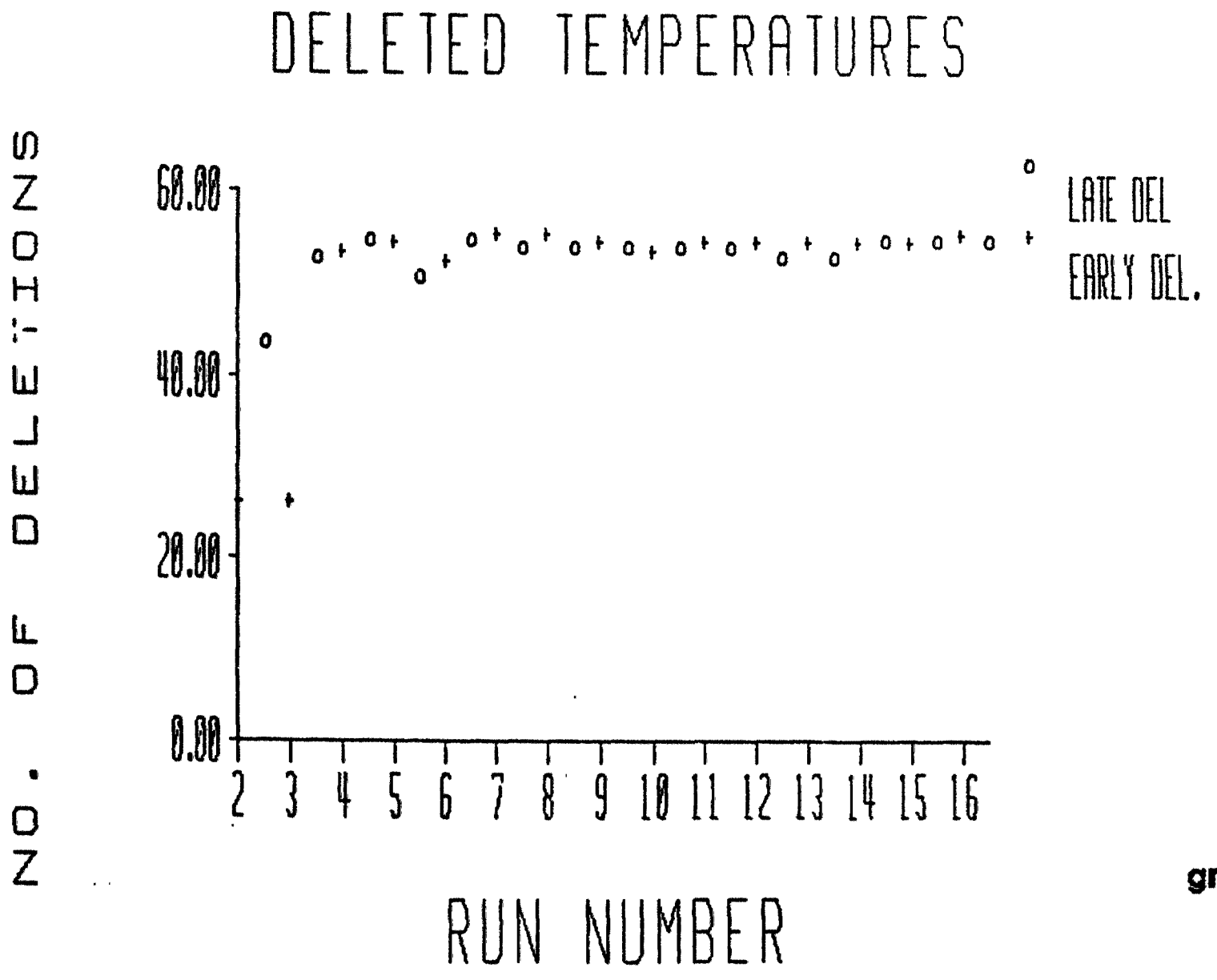
group 2


TOTAL DELETED HEADERS
NO. OF DELETIONS
150.00
100.00
50.00
0.00
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
RUN NUMBER
LATE DEL
EARLY DEL

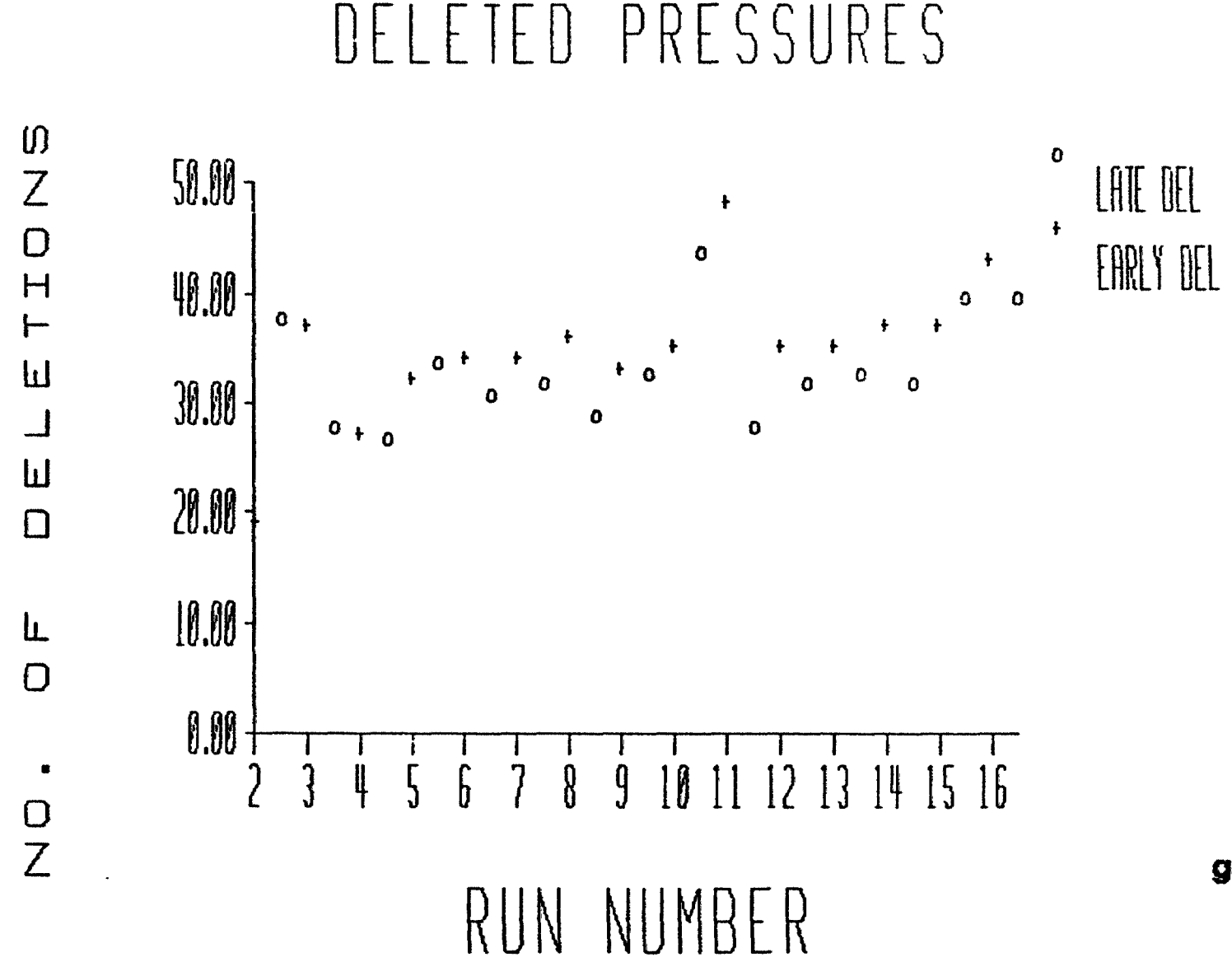
group 2


DELETED M,L,&D HEADERS
NO. OF DELETIONS
25.00
20.00
15.00
10.00
5.00
0.00
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
RUN NUMBER
LATE DEL
EARLY DEL

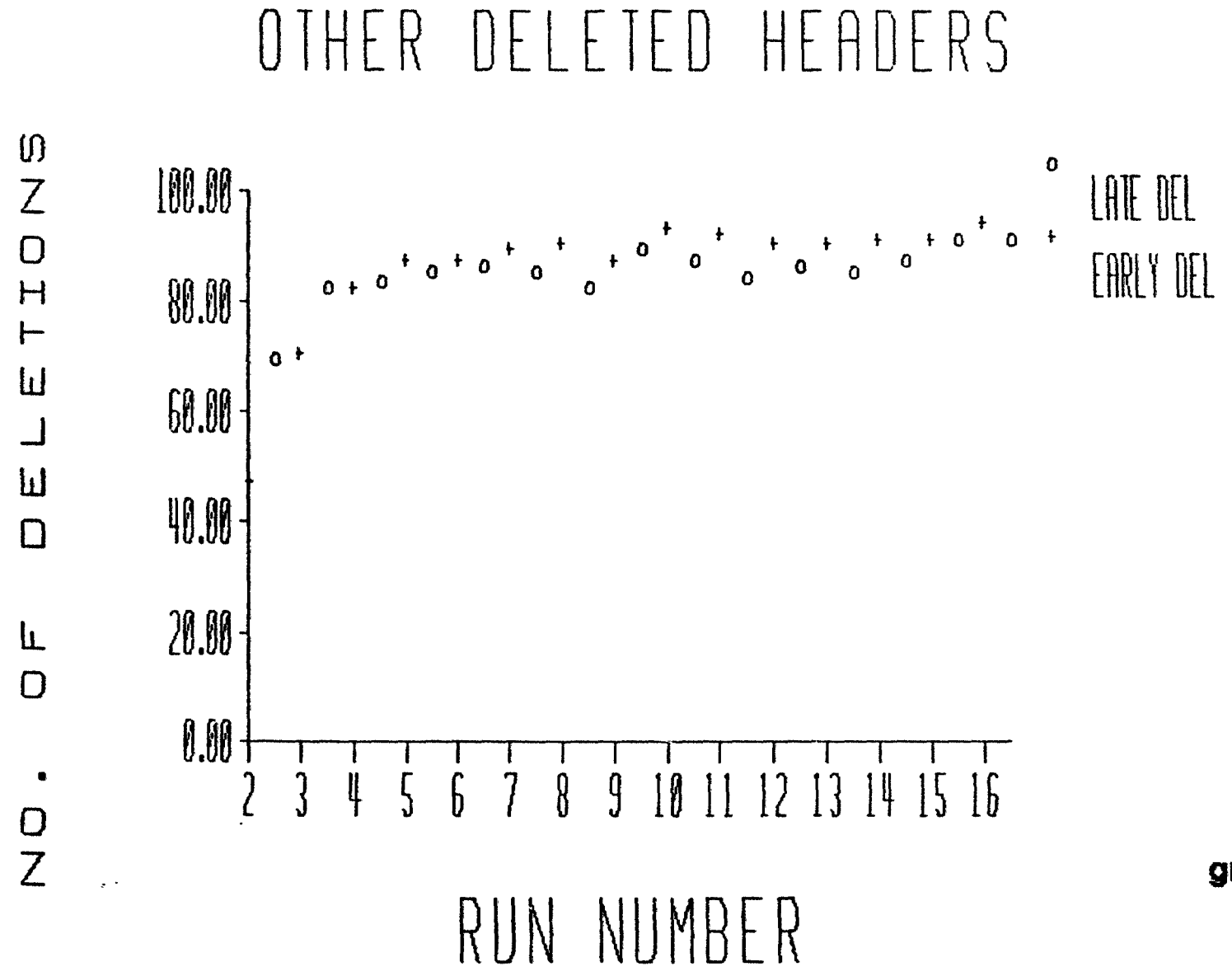
group 2


DELETED TEMPERATURES
NO. OF DELETIONS
60.00
40.00
20.00
0.00
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
RUN NUMBER
LATE DEL
EARLY DEL.

group 2

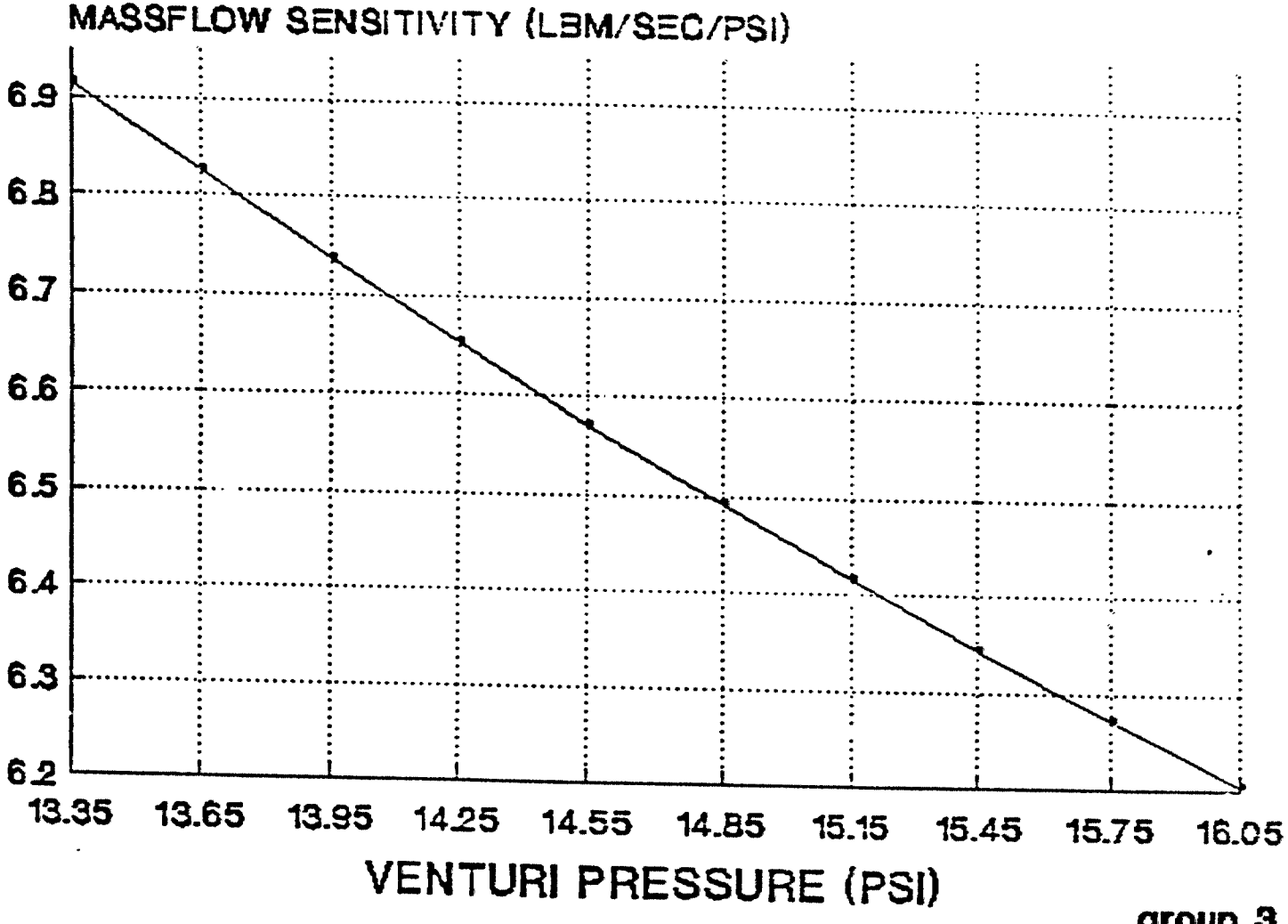

DELETED PRESSURES
NO. OF DELETIONS
50.00
40.00
30.00
20.00
10.00
0.00
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
RUN NUMBER
LATE DEL
EARLY DEL

group 2


OTHER DELETED HEADERS
NO. OF DELETIONS
100.00
80.00
60.00
40.00
20.00
0.00
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
RUN NUMBER
LATE DEL
EARLY DEL

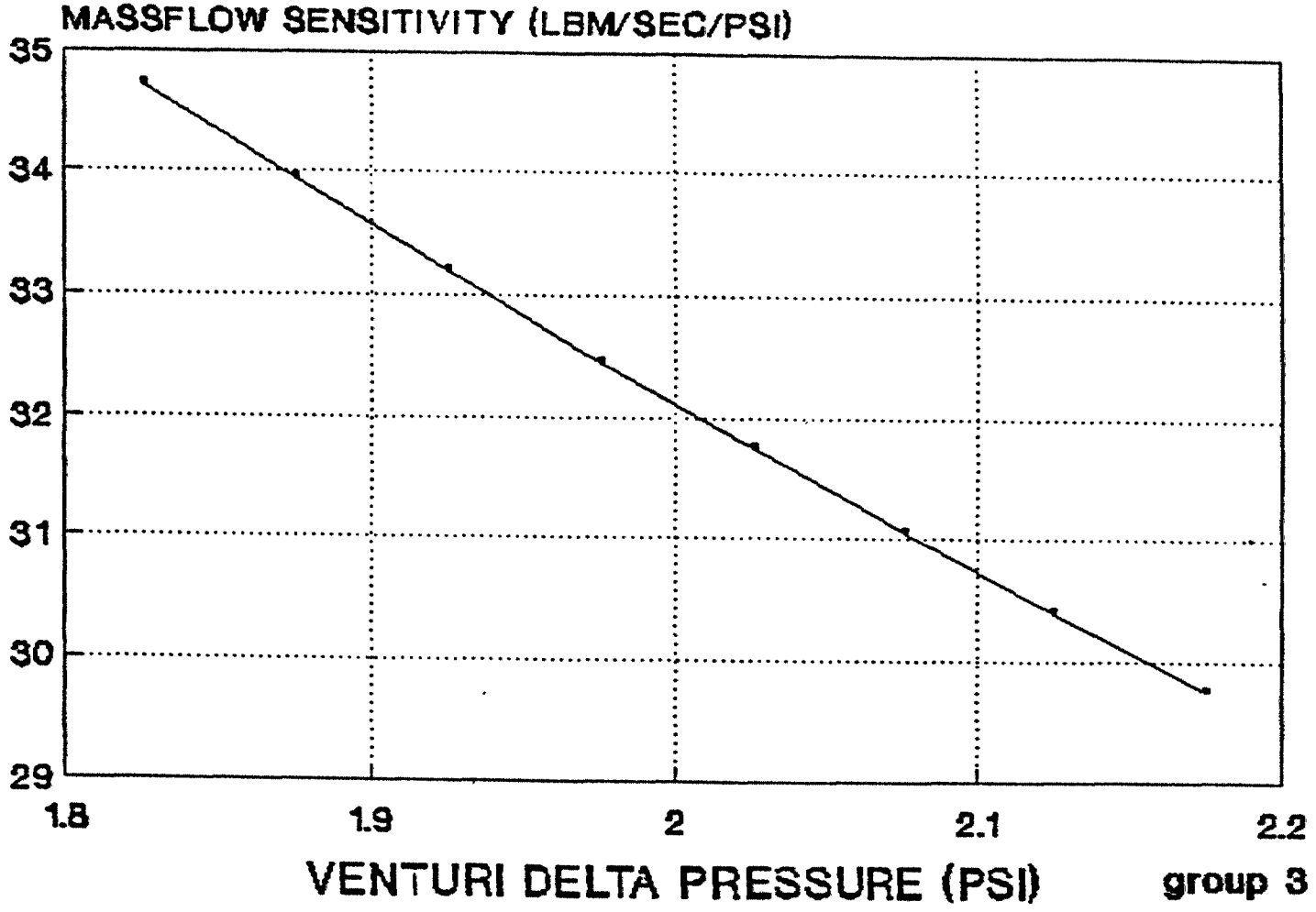
group 2

SENSITIVITY OF WAVT TO PSV

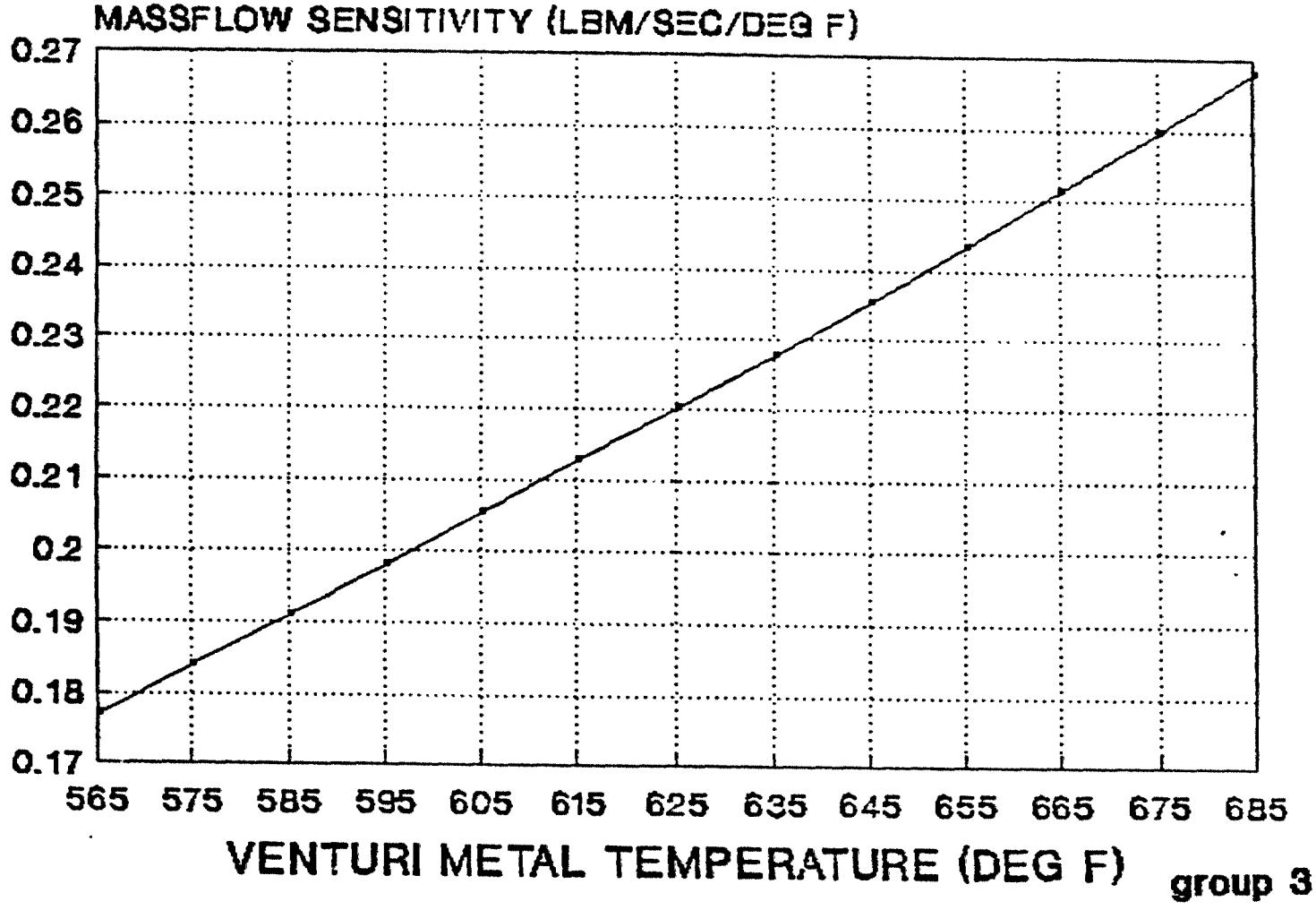


group 3

SENSITIVITY OF WAVT TO PDV

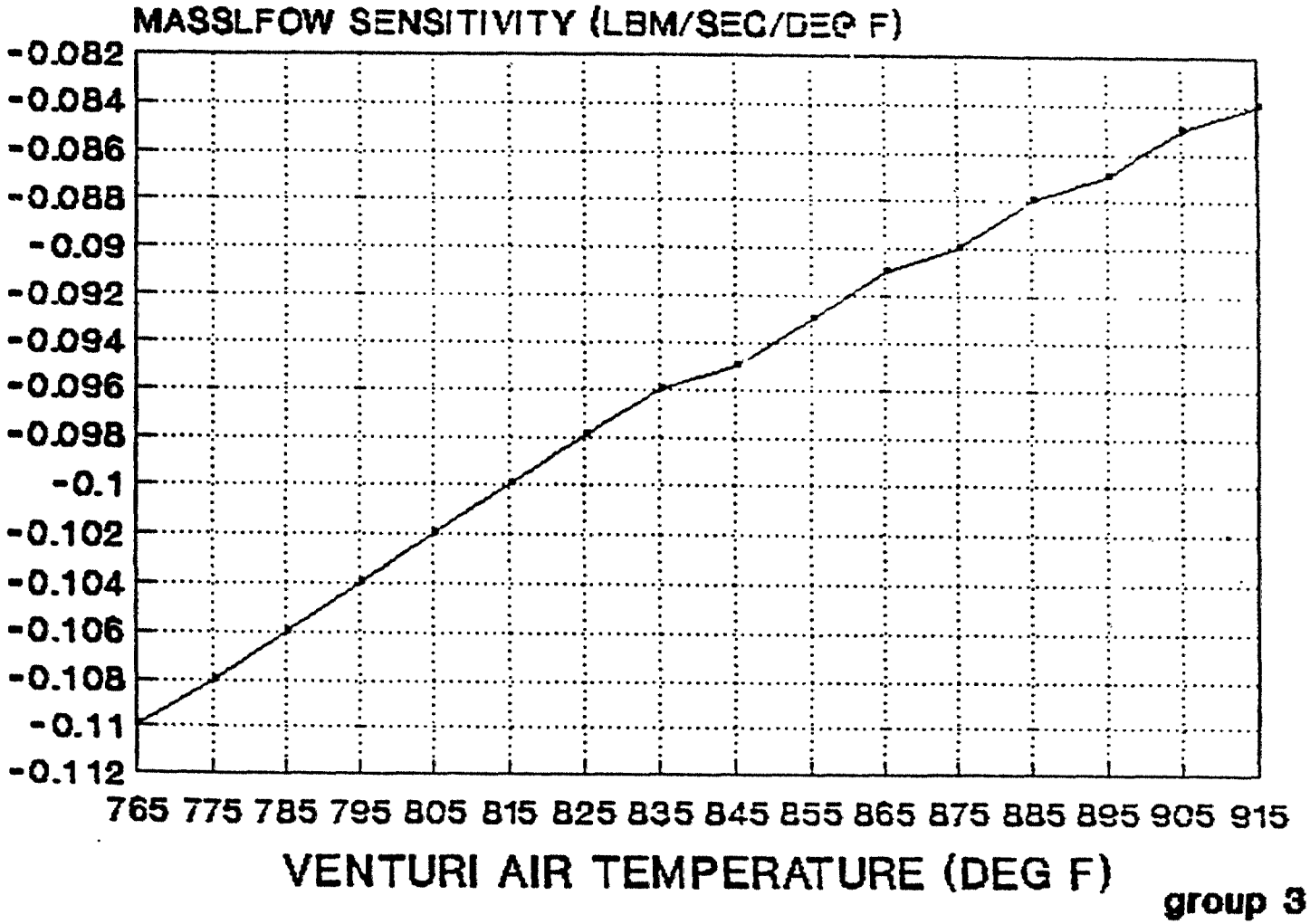


SENSITIVITY OF WAVT TO TMV



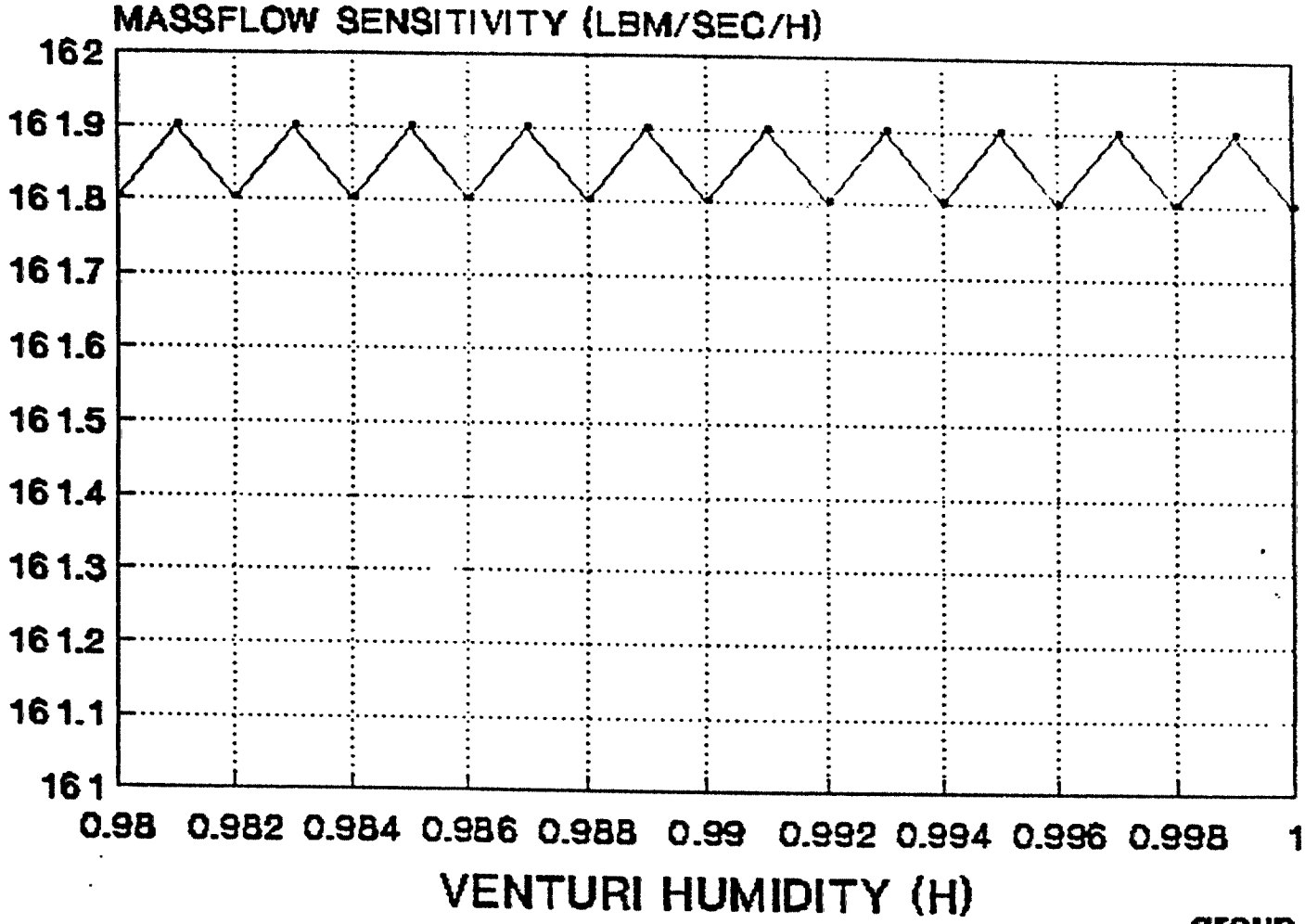
group 3

SENSITIVITY OF WAVT TO TTV



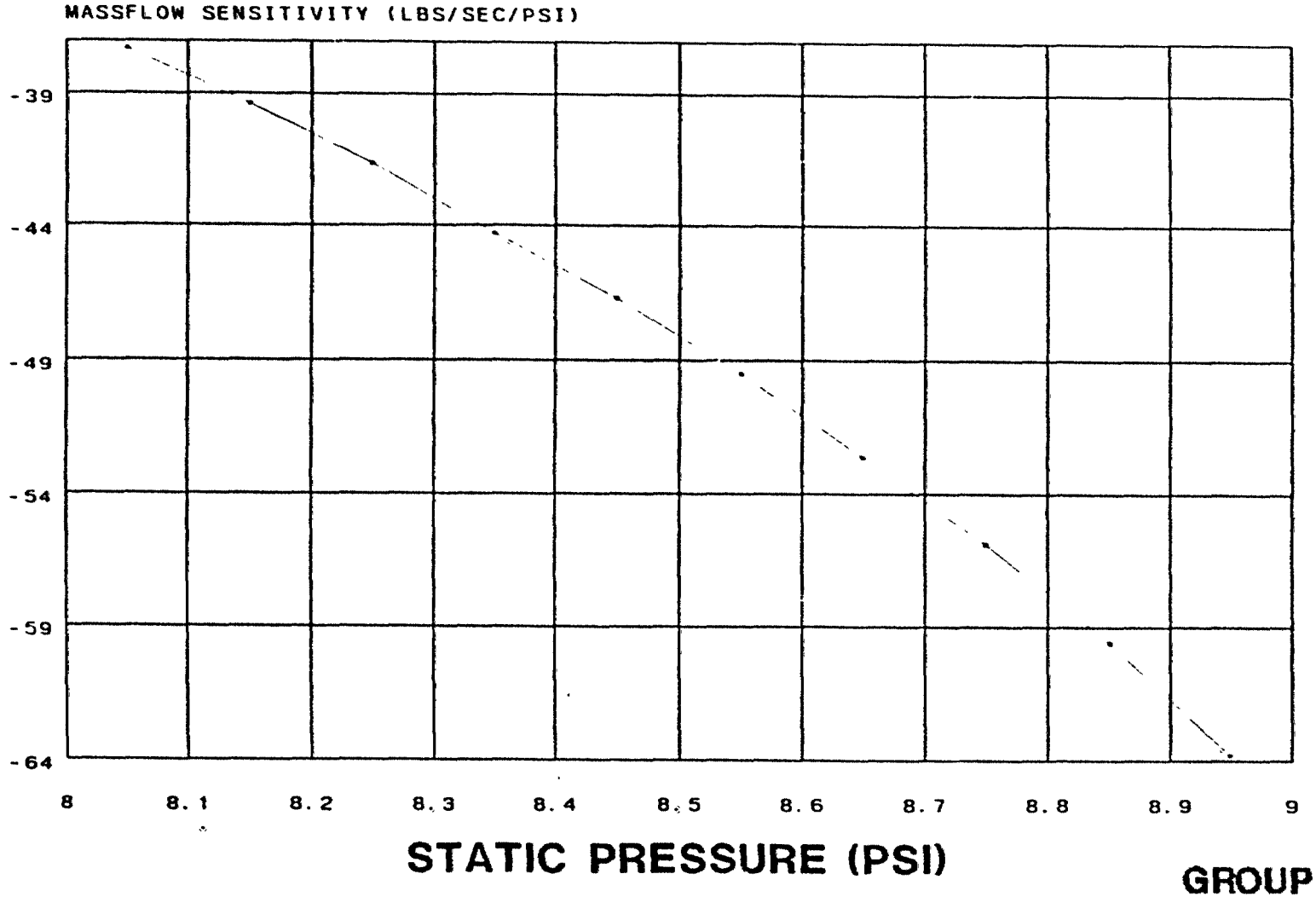
group 3

SENSITIVITY OF WAVT TO HUMIDITY



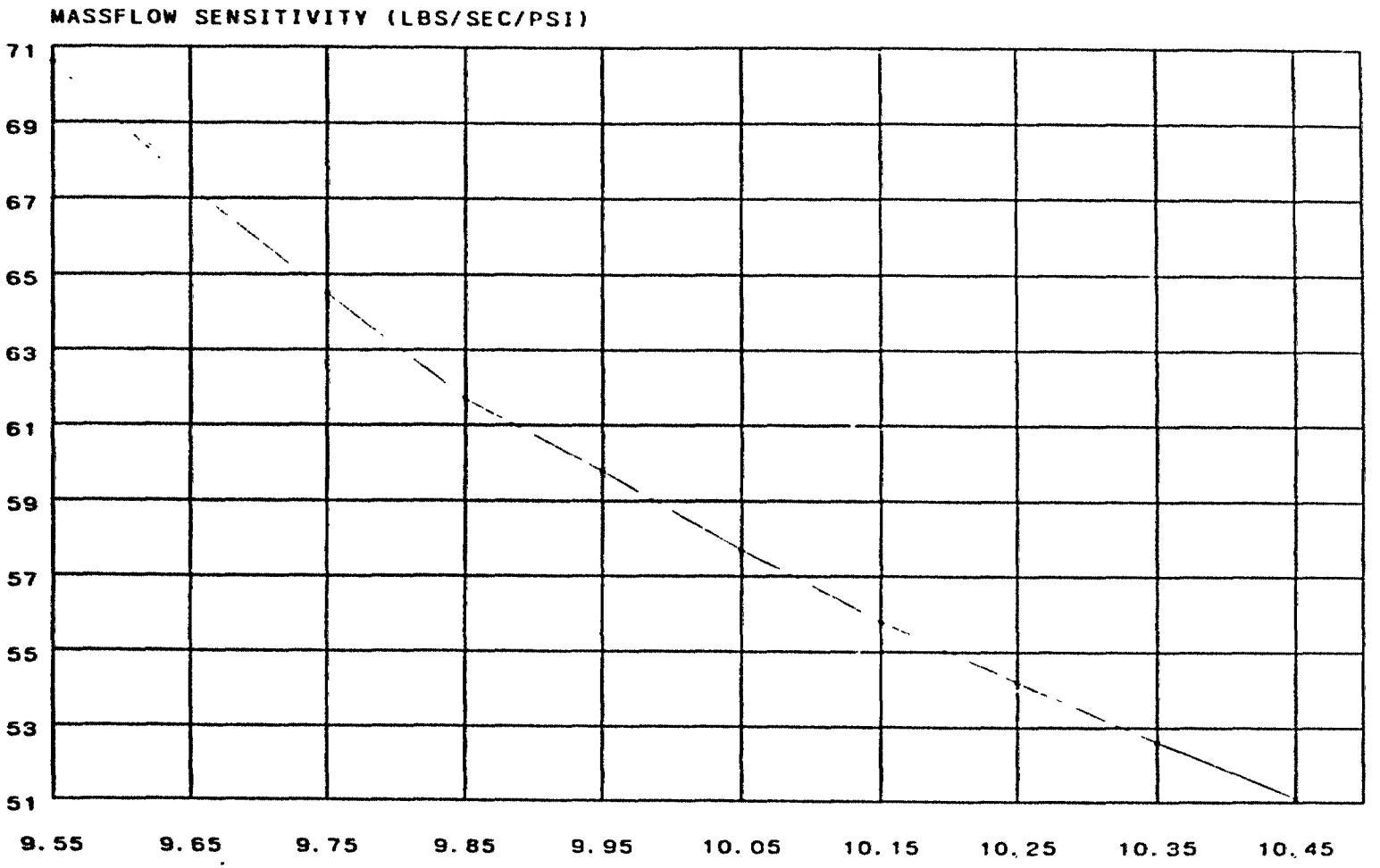
group 3

BELLMOUTH MASSFLOW TO PSB



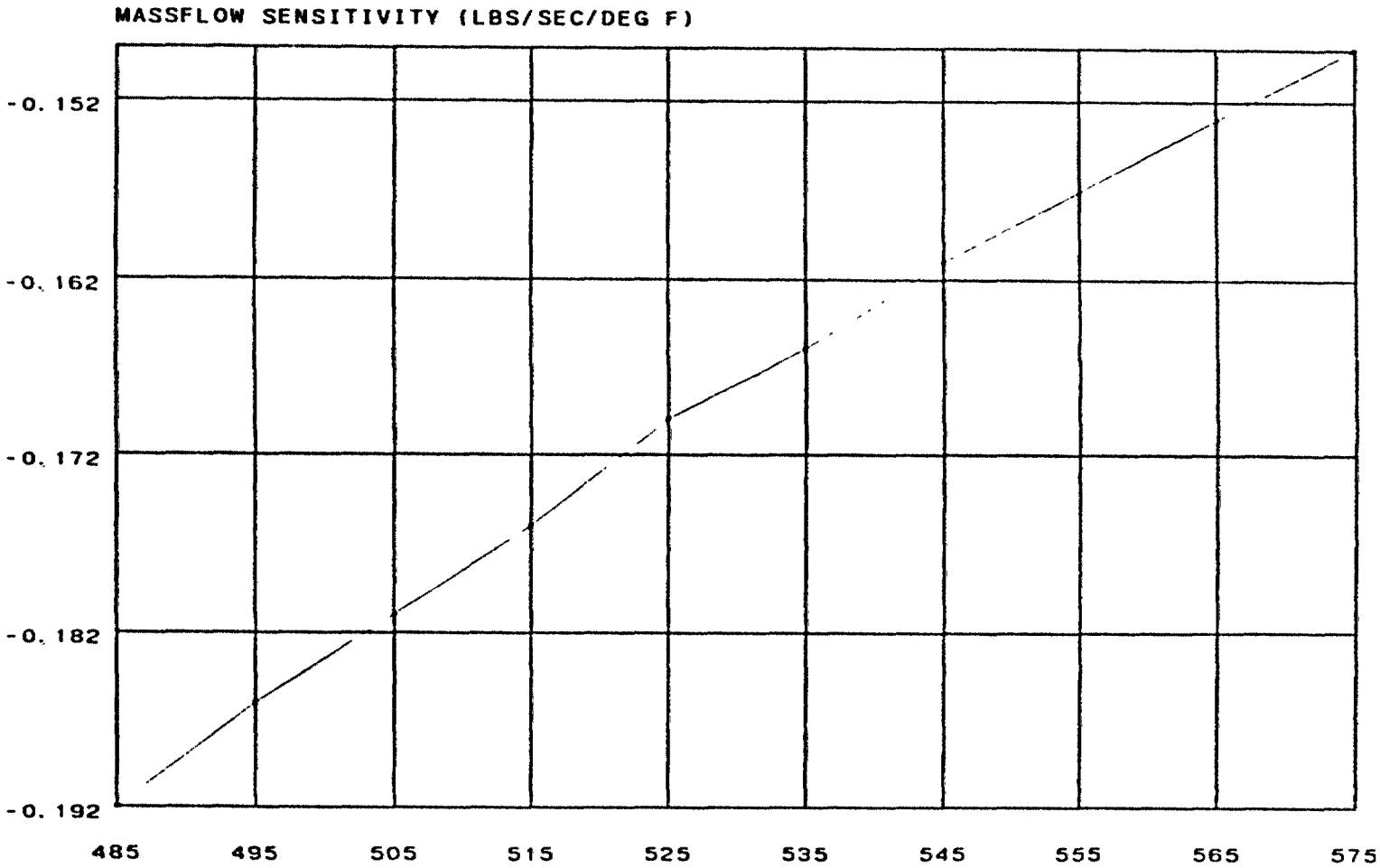
GROUP 3

BELLMOUTH MASSFLOW TO PTB



GROUP 3

BELLMOUTH MASSFLOW TO TTB



GROUP 3

"FROZEN START UP"

CAROL ROGERS
MENTOR: BRIAN G. HAGER

AUGUST 13, 1990

In conjunction with the Air Force High School Apprenticeship Program and Universal Energy Systems, I was assigned to the project, "Frozen Start Up" which was being conducted in the thermal division of the Aero Propulsion Laboratory. "Frozen Start Up" is an experiment which involves heat pipes.

Heat pipes are designed to transport large amounts of heat from one location to another over a small temperature gradient with a small unit size. The design and operating principles of the heat pipe are simple. It is a closed tube or chamber whose inner surface is lined with a porous capillary wick. The wick is saturated with a liquid working fluid. Heat is then applied to the evaporator end of the pipe, thus allowing the liquid to vaporize. The vapor travels through the vapor core to the condenser section where it condenses releasing the latent heat of vaporization. A capillary pressure is created by the depletion of liquid in the wick, causing the liquid to travel from the condenser to the evaporator. The heat pipe should continually transport the latent heat of vaporization from the evaporator to the condenser without drying out the wick.

One practical application of the heat pipe is for thermal control in spacecraft. Realizing that in space the heat pipe must operate in sub-zero temperatures, research is currently being done in the frozen state to simulate the conditions in space. However, there are several problems associated with frozen heat pipes. The most serious of these is frozen start up. A frozen pipe usually dries out before it becomes fully operational. The solid liquifies and eventually vaporizes in the evaporator section of the pipe.

However, the working fluid often remains in its solid state throughout the middle and condenser section preventing the condensed vapor from traveling up to the evaporator. Thus, dry out occurs.

The purpose for this "frozen start up" experiment is to add to the general research data available. Similar tests have been conducted by W. Bowman of the United States Air Force Academy providing results showing a successful "frozen start up."

The first of the procedures followed were calculations. These calculations were used to determine the point at which the pipe would dry out. Taking into account the characteristics of the pipe, the information in the following equations were used to find the capillary, sonic, boiling, and entrainment limits, respectively.

$$Q_{c_{max}} = \frac{(QL)_{c_{max}}}{(0.5L_c+L_a+0.5L_e)}$$

$$Q_{s_{max}} = A_v \rho_v \lambda_v \sqrt{\left[\frac{\gamma_v R_v T_v}{2(\gamma_v+1)}\right]}$$

$$Q_{b_{max}} = \frac{2\pi L_e K_e T_v}{\lambda \rho_v \ln(r_1/r_v)}\left(\frac{2\sigma}{r_n}-P_c\right)$$

$$Q_{e_{max}} = A_v \lambda \sqrt{\left(\frac{\sigma \rho_v}{2 r_{h_e}}\right)}$$

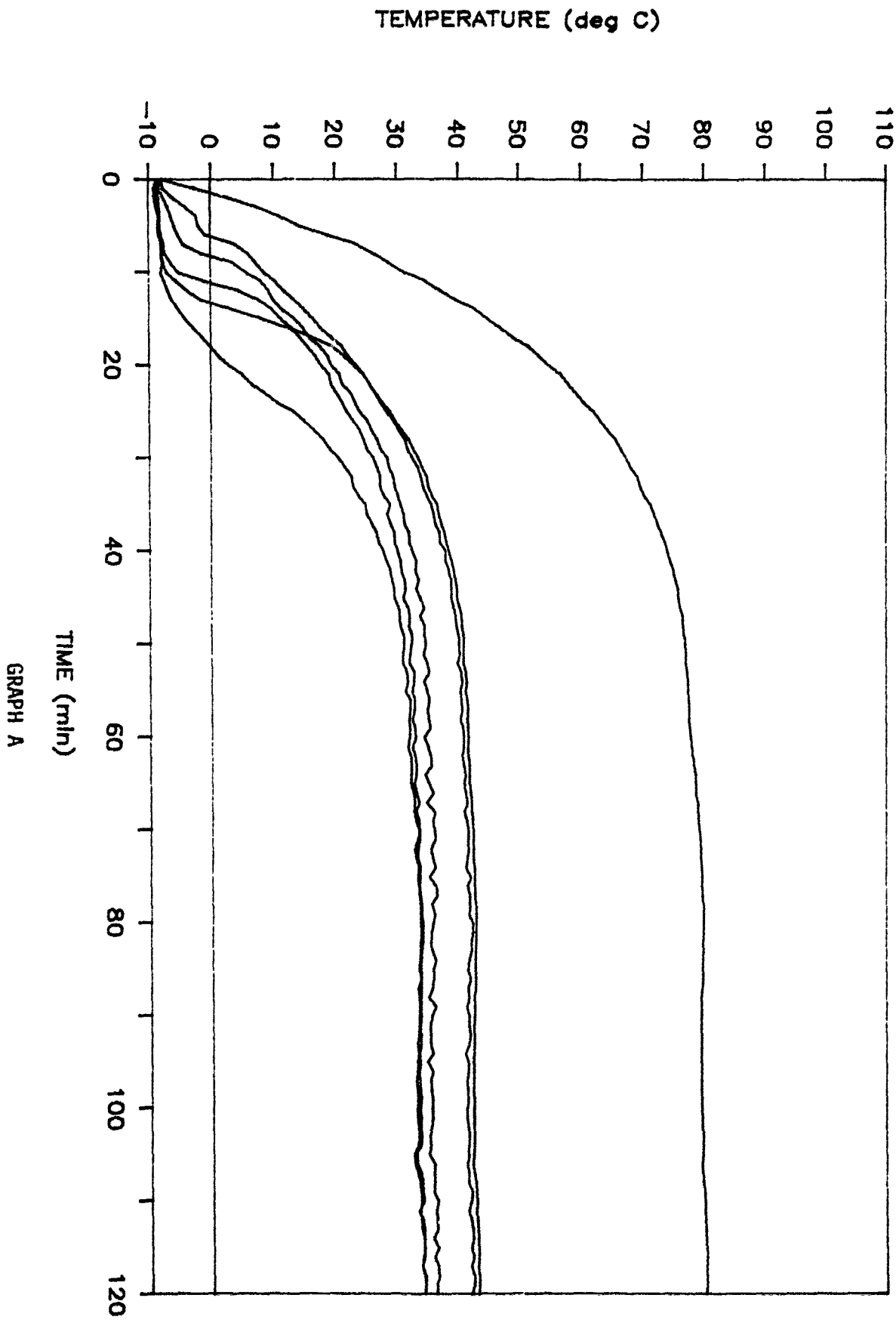
The next step involved the instrumentation of the pipe. The copper pipe was 24 inches long and had a diameter of one-half inch. The wick inside the pipe was a copper wire screen with the working fluid being water. Type T thermocouples were placed every two inches along the pipe in order to read the temperatures of the pipe. These thermocouples were then connected to a data logging system, Fluke 2286A. After programming the Fluke to record temperatures at given time intervals, tests were started. The first tests were characterizations with different amounts of power supplied to the heat source. The heat source was a clam shell heater capable reaching temperatures above 600 degrees Celsius. Characterization tests were run to provide data in the form of graphs for comparison to "frozen start up" tests.

When all the characterizations of the pipe were completed, "frozen start up" tests were implemented. Cooling the environmental chamber at -15 degrees Celsius, 50 watts and 100 watts of power were added to the heater to begin the process of "frozen start up." Other tests were conducted with the chamber at -25 degrees Celsius and -65 degrees Celsius. After analyzing the results of the tests, it became apparent that the heater was not generating the proper amount of heat for its wattage due to the fact that most of its power was put into heating its own mass. Therefore, it was decided to replace the clam shell heater with a

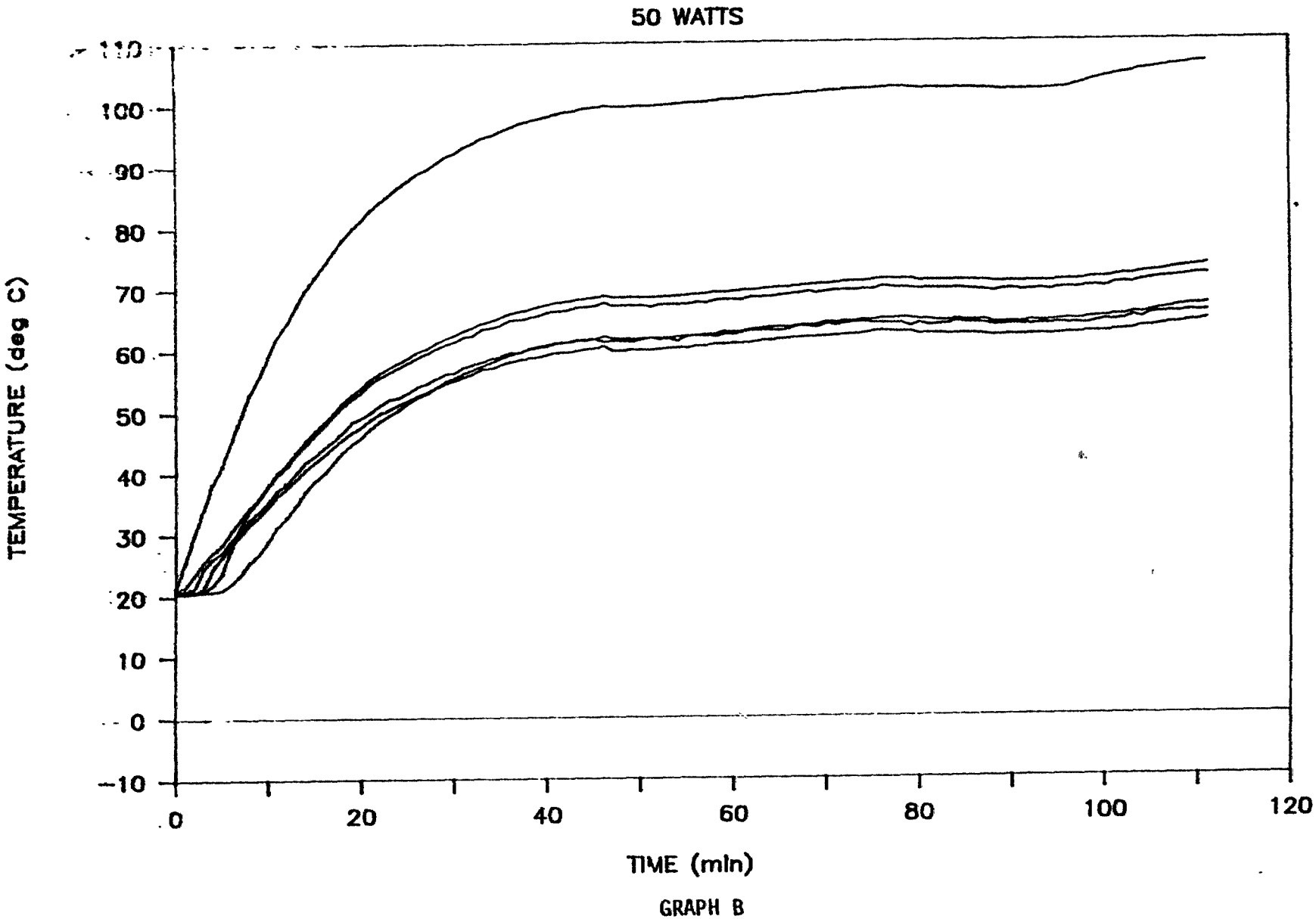
smaller, more efficient one. Again, characterizations were run for comparison using the smaller heater.

With a new temperature controller, the chamber was cooled to a constant -10 degree Celsius. To the frozen pipe, different amounts of power were added to conduct "frozen start up" tests at 50 watts and 100 watts. By comparing the graphs of these tests to the characterizations, it was concluded that successful "frozen start ups" have been achieved (See graph A and B).

This experiment has provided valuable data for frozen heat pipes. Further tests could be done with different working fluids and/or wick types. This will eventually make heat pipes vital to science and technology in space.


TEMPERATURE (deg C)
-10
0
10
20
30
40
50
60
70
80
90
100
110
0
20
40
60
80
100
120
TIME (min)
FSU AT −10 DEG C
50 WATTS
GRAPH A

CHARACTERIZATION OF HEAT PIPE



GRAPH B

BIBLIOGRAPHY

Chi, S.W. Heat Pipe Theory and Practice, New York: Mcgraw-Hill Book Company, 1976, pp. 1-94.

Dunn, P.D. and Reay, D.A. Heat Pipes, Oxford: Pergamon Press, 1982, pp. 1-10.

Acknowledgements

This research was sponsored by the Aero Propulsion and Power Laboratory at the Air Force Wright Research and Development Center. I would like to thank the following people for their guidance and assistance with the "Frozen Start Up" project: Brian Hager, Dr. Won Soon Chang, Mike Ryan, Don Brigner, John Tennant, Joe Mantle, and JoElla Pinckney.

ARMAMENT LABORATORY

WIND VELOCIMETERS FOR CALCULATING BALLISTIC TRAJECTORIES

STEVEN BRYAN
CRESTVIEW HIGH SCHOOL

FUZES AND GUNS BRANCH
MENTOR: DON CUNARD
AUGUST 15,1990

INTRODUCTION
SECTION 1

This summer I worked in the Fuzes and Guns Branch of the Air Force Armament Laboratory at Eglin Air Force Base. I helped my mentor, Don Cunard, with a exploratory development fire control program called BSTING (Beam Sight Technology Incorporating Night vision Goggles). BSTING is a laser targeting system used in conjunction with night vision goggles which will help Special Operations helicopter gunners be more effective.

My assignment was to calibrate and integrate instrumentation for BSTING flight testing at China Lake Naval Weapons Center, CA, and to research wind measurement systems to enhance BSTING's capabilities.

I worked with the machines in Bay 9 to construct parts for the BSTING test. I also researched three wind systems which I will describe in detail later. By talking with MNF technicians I learned the importance of mechanical safeties in fuzes as well.

ACKNOWLEDGMENTS
SECTION 2

I would like to thank my mentor Don Cunard for allowing me to return again this summer and for explaining the BSTING program to me. I would like to thank Robert Orgusaar for explaining the optical systems that I researched more fully to me. I also would like to extend thanks to Shelly Pollard and Mike Caraway for helping me with the computer programs. Fred Bath and Scott Turner also deserve recognition for their guidance in showing me how to use the machines in the bay and in constructing BSTING components. I have enjoyed my summer here at the MNF branch and I would like to thank all the other people at my branch for their help and support.

BACKGROUND
SECTION 3

My research project was related to the BSTING program. Further information on the BSTING program is needed to fully understand the reason behind my research. The systems objective is to increase the effectiveness of special operations helicopter gunners that use crew operated weapons and night vision goggles. This will be done by replacing tracer rounds with a laser gun sight that will increase ammunition efficiency, increase first burst hits, increase aircraft survivability, and decrease training time. The BSTING uses an infra-red laser that is offset from a gun's boreline with two servomotors to compensate for azimuth and elevation. Currently the system does not take wind velocity in to account along the flight path of the bullet nor at the target's location. This is where my research fit in.

PROJECT
SECTION 4

I researched three types of wind measurement systems; two systems using carbon dioxide lasers operating at 10.6 micrometers, the other used a neodymium:yag laser operating at 1.06 micrometers.

The first system I investigated was the speckle-turbulence interaction system. This system used a coherent, continuous wave carbon dioxide laser. The laser beam was split in a 90% / 10% ratio. The 90% beam was then modulated to 37.5 MHz and transmitted into the atmosphere. The remaining 10% of the beam acted as the local oscillator and was modulated at 42.5 MHz.

The system had 2 receivers in a horizontal plane, each with a separate field-of-view. As the return signals are received they are mixed with the local oscillator signal to cancel out and leave a 5 MHz signal. This was done to eliminate signal noise. The 5 MHz signal was then further filtered down to 100 kHz and precision rectified. An average peak value was measured for each receiver. As the wind blew the particles in a certain direction the peak values would slowly transfer from one receiver to the other. The time delay for the transfer and the distance between the receivers was measured and used to calculate the wind velocity. This system had an error of .5 meters per second for a range of 1 km.

(SEE FIG.1)

The neodymiun:yag laser was used to measure wind velocity in the second system I researched. This system was developed by Coherent Technologies in Boulder, Colorado. The nd:yag is diode pumped and sent through an isolator to make the beam coherent. The beam is then split, one part becoming the local oscillator and the other the transmit beam.

The transmit beam is modulated at 200 MHz and then gated to create a pulsed effect. The beam is then sent through a amplifier and telescoped into the atmosphere. The same telescope collects the return signal and sends it to a fiber optic coupler where the return signal is mixed with the local oscillator signal. A cooled detector is unnecessary since a laser diode is used, making the system less expensive. The mixed signal is sent from the detector to a signal processor where the doppler shift is measured and correlated to the wind velocity. This system has a range of 3.5 km with a resolution of 200 m.

(SEE FIG.2)

The last system I researched was a coherent doppler lidar which used a carbon dioxide laser. The beam was split and pulsed with one portion being the transmit beam and the other being the local oscillator. The beam was telescoped to the target and backscatter returns from aerosol particles was received by the telescope. From there the signal is sent to a photomixing receiver and coupled with the local

oscillator signal. The receiver changes the optical frequencies to radio frequencies, a process known as heterodyne detection. Signal processing is then used to find the wind velocity.

The spectral width of the transmit beam is important. If it is not wide enough to overlap the largest expected doppler shifted signal there will be no cancellation of frequencies, no doppler shift, and hence, the wind velocity cannot be calculated. This system has a range of up to 30 km with a resolution between 150 and 300 meters.

(SEE FIG.3)

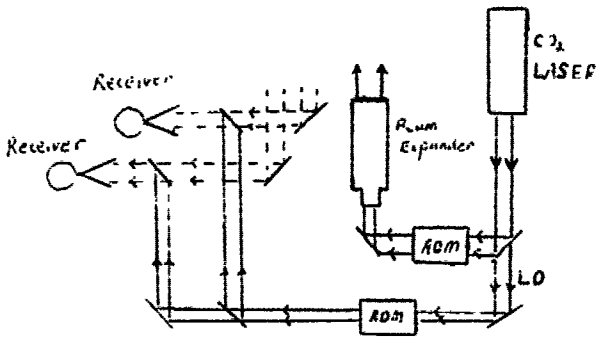


FIG. 1

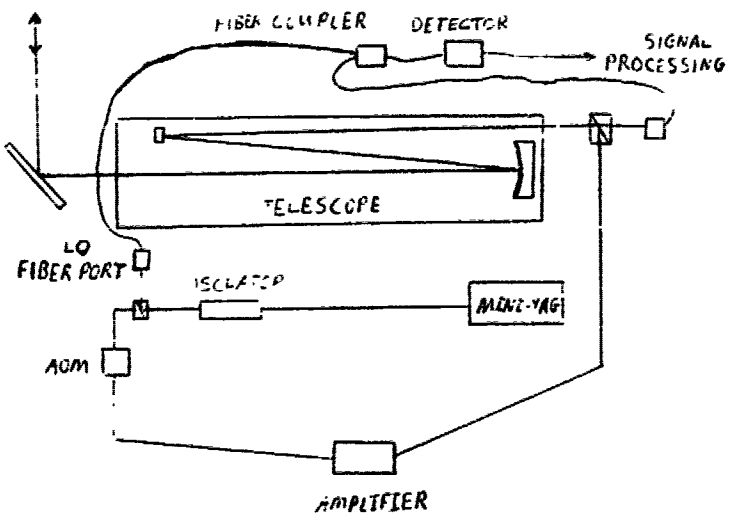


FIG 2

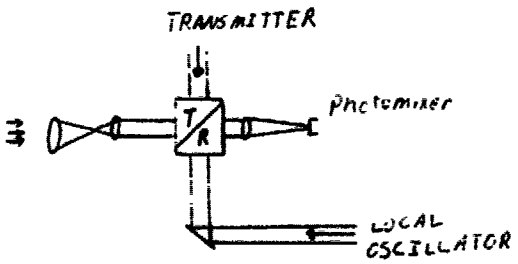


FIG. 3

CONCLUSION
SECTION 5

In conclusion, all the systems I researched had the necessary range and resolution to upgrade the BSTING program but the most likely candidates use laser diodes and uncooled detectors. The neodtmium:yag and the speckle-turbulence systems fit these specifications.

The nd:yag system has both laser diodes and uncooled detectors. But a disadvantage for this system is that it has a telescope. The BSTING program needs the gun sight and related equipment to be as compact as possible and a telescope would be bulky.

The speckle-turbulence interaction system has a potential application to BSTING. Although it currently uses a carbon dioxide laser and cooled detectors, the low power required to operated the system (1 to 2 watts) would allow laser diodes and uncooled detectors to be substituted. A strength of this system is that it has no telescope. A disadvantage of this system is that wind speed measurements are only possible in the horizontal plane because of the horizontal orientation of the detectors; a third, vertically oriented, detector would need to be added to account for vertical wind movement.

MISCELLANEOUS
SECTION 6

This summer I also improved my skill on the GEM graphics and WordStar programs in making my presentation. I also learned how to read wiring diagrams and maintenance schematics in calibrating a magnetic recorder to be used in BSTING flight tests at China Lake Naval Weapons Center, CA.

In discussions with Nunzio Zummo, our Laboratory Section Chief, I learned the importance of fuze safe-and-arm mechanisms for explosive munitions. Safe-and-arm mechanisms ensure that the detonator is not aligned with or is kept separate from the main explosive charge of the munition. This is important so that if the munition were in a fire (either on-board an aircraft or in storage) or inadvertantly dropped, the main munition charge will not accidently detonate. Examples of weapons that require fuzes are: high explosive incendiary gun ammunition, missiles, bombs, mines, etc.

REFERENCE PAGE

Applied Optics, vol.27 no.12, "REMOTE SENSING OF ATMOSPHERIC WINDS USING SPECKLE-TURBULENCE INTERACTION", Optical Society of America, New York, NY, June 1988, p.2532-2538.

Don Cunard, Mechanical Engineer, Fuzes and Guns Branch, Munitions Division, interview. July 1990.

Laser Focus World, vol.25 no.6, "Solid-state lidar development promises longer lifetimes", PennWell Publishing Company, Tulsa, OK, June 1989, p.23-24.

Nunzio Zummo, Laboratory Section Chief, Fuzes and Guns Branch, Munitions Division, interview. July-August 1990.

Proceedings of the IEEE, vol.77 no.3, "Coherent Doppler Lidar For Measurement of Wind Fields", Institute of Electrical and Electronics Engineers, INC., New York, NY, March 1989, p.449-461.

Robert Orgusaar, Electrical Engineer, Fuzes and Guns Branch, Munitions Division, interview. July-August 1990.

S.A.S.S.E.

Written by :
Nicki Cook

Mentor:
Tim Poth

AFATL/SAI

August 13, 1990

TABLE OF APPENDICES

A. MAXMAG500.FOR

B. Alpha

C. Beta

D. Alpha Output

E. Beta Output

F. D.U.M.B.O.

G. Wow

H. Holman Spectral Irradiance

I. Number of Stars vs. Wavelength

J. Number of Stars vs. N.E.F.D.

My project for the summer of 1990 was entitled S.A.S.S.E. (Star Availability for Sensor - Specific Evaluation). This program had two basic parts: Alpha/Beta and D.U.M.B.O.

Part one consisted of two programs: Alpha (which requests the user to input star names and addressed to be matched with SAO identification numbers in order to modify existing star map data files) and Beta (which places the brightest 500 stars alphabetically in a file, grouping stars by constellation).

Part two of S.A.S.S.E. was appropriately entitled D.U.M.B.O. (Determining Uranographic Matches from Bands Observed). It matched stars in an SAO visual map and an AFGL IR map and then computed spectral irradiance in a waveband of interest.

Both the Alpha/Beta and D.U.M.B.O. portions of S.A.S.S.E. enabled much sensor - specific evaluation to be performed during the summer at AFATL/SAI.

I would like to give a special "thanks" to all of the people who allowed me to participate in the HSAP program again this year, especially Mr. Don Harrison for taking on and fulfilling the responsibility of making the program run smoothly once again

Also, I want to thank everyone at SAI for putting up with me (and not throwing my shoes away when I wasn't looking), especially Steve Perillo for his patience and helpful hints when my masterminded mentor was gone.

I want to extend a sincere "thank you" to Randy Wells, our SETA at SAIC, simply for being a physicist and unintentionally convincing me that there is still hope for me in a scientific field. Randy, know that all of your input and ideas helped substantially in making the program such a valuable learning experience for me.

Finally, last but not least, I want to say "thank you" from the bottom of my heart to Capt. Tim Poth for all of his many hours of explanations and his master-minded touch at debugging. More than anything else, Tim, I thank you for being a true friend, allowing my last summer job in the "teen-age world" to be one paired with enjoyable memories and experiences.

During the summer of 1989, much work on stellar navigation was done. This work, though, was neither sensor specific nor allowed for a passive sensor. It did however, research an active sensor that would sight and evaluate a single star, reorient the sensor, sight a second star, and determine position based on those two measurements.

This summer a new set of criteria was established: alterations for a passive sensor needed to be performed and stars needed to be identified by constellation names and star addresses verses simply star numbers or positions.

There were two maps used to research and evaluate star sensing under this new criteria: an SAO visible map of 258,997 stars and an AFGL IR map of only 6,146 stars.

Before the Alpha/Beta programs were written, some hands on astronomy was done. Five hundred stars were located on star maps and their names and addressed were recorded. Next, programs Alpha and beta were written to take this information and convert it to star data files with other star data on them.

Next, D.U.M.B.O. had been written by engineer Laura Sterrett, during the spring of 1990, based on another program from Rockwell International. Laura was soon relocated in D.C. before debugging and the final data manipulations were incorporated into the program.

Then, for the summer of 1990's secondary project, D.U.M.B.O. (Determining Uranographic Matches from Bands Observed) had to be completed and perfected. After becoming well acquainted with Rockwell's algorithms and spectral irradiance conversions, D.U.M.B.O. was successfully completed.

Both of these portions of S.A.S.S.E. (Star Availability for Sensor - Specific Evaluation) enabled observation and further understanding of a passive star sensor, its applications, and its limitations.

After completing stellar data files and confirming their accuracy, these files can be accessed by the star sensor for periodic navigational checks during flight to aide the interceptor. These checks provide assurance of the interceptor's mid-course performance to confirm final interception of target.

These star data files will be accessed by the star sensor and the appropriate data will be fed into the Kalman Filter for further necessary computations.

To perform the necessary star location on star maps, right ascension must be converted to degrees by multiplying by the conversion factor "2/3". Next, the right ascension and declination are read on a celestial map much like latitude and longitude, respectively. These star ID's consist of a Greek letter to identify which star in the constellation is referred to, the constellation name, and proper name which may or may not be known. From these addresses, the stars were individually identified and recorded in a catalog with other data such as visual magnitude and spectral type.

After this, the brightest 500 stars identified through the program MAXMAG500.FOR (Appendix A) were combined with the manually determined star addressed and compiled, using the Alpha (Appendix B) and Beta (Appendix C) programs. The Alpha program, in turn, organized the brightest 500 stars alphabetically by constellation and descending visual magnitudes. Then, the Beta program organizes this data file according to constellation groupings and descending visual magnitudes.

The second portion of S.A.S.S.E. was entitled D.U.M.B.O. (Determining Uranographic Matches from Bands Observed). After reviewing the skeletal program for stellar matching, many alterations and additions were made. Each subroutine was checked for errors in logic and typing. The subroutine IRRADIANCE was then added to convert spectral irradiance into specific wavebands. After completing the modifications on D.U.M.B.O. (Appendix F), the program enabled the computer to read in the SAO visual and AFGL IR maps and make to data matches to insure reliability on consistent star data. The first match checked to positions on the two maps and matched each star within a .02 microns error band. Next, these stars were stored in a file, and the second check was performed. This checked compared visual

magnitude to assure its accuracy within two magnitudes of each other. Next, from the spectral type, temperature was determined from the Holman Heat Transfer book (Appendix H).

Then, many spectral irradiance calculations were performed. First, the peak spectral irradiance was computed:

$$H_{\lambda PEAK} = H_{OPEAK} * \left[10^{(-VMAG/2.5)}\right] \qquad (1)$$

where

H_{OPEAK} = peak spectral irradiance of a zero visual magnitude star of effective temperature T

Next, the spectral irradiance was estimated at the appropriate wavelength by evaluating the blackbody curve of peak spectral irradiance:

$$H'(\lambda) = \left(H_{\lambda PEAK} * x^5\right) / \left(c * \left(e^x - 1\right)\right) \qquad (2)$$

where

c = 21.199

x = 14387.9 $/\mu m * K / \lambda * T$

Following that, the spectral irradiance magnitude, M', had to be solved using:

$$M' = -2.5 \log\left[H'(\lambda) / H_0(\lambda)\right] \qquad (3)$$

where

H_o (4.2 μm) = 3.6×10^{-15} Watts/cm^2 *μm

H_o (11.0 μm) = 8.7×10^{-17} Watts/cm^2 *μm

H_o (20.0 μm) = 8.2×10^{-18} Watts/cm^2 *μm

H_o (27.0 μm) = 2.5×10^{-18} Watts/cm^2 *μm

The spectral irradiance was then recalculated by integrating equation (2):

$$SR = \int_o^\infty (H_{\lambda peak} * x^5) / (C * (e^x - 1)) \, d\lambda \qquad (4)$$

The integral equation (5) over all wavelengths is the blackbody radiation law:

$$E_{b\lambda} = (C_1 \cdot \lambda^{-5}) / (e^{C_2/\lambda \cdot T} - 1) \qquad (5)$$

where

$C_1 = 3.743 \times 10^8$ Watts*μm^4/m^2
$C_2 = 14387.9$ μm*K

Since equations (2) and (5) are similar and differ only by a constant, integration of equation (4) could be done by multiplying equation (6) by this constant. Introducing:

$$E_{bo\text{-}\lambda} = \sigma T^4 \qquad (6)$$

σ = Planck's constant, 5.669×10^{-12} Watts/cm^2*K^4

$$\frac{H_{\lambda peak} \cdot x^5}{c(e^x - 1)} = Z \frac{C_1}{\lambda^5 (e^x - 1)} \qquad (7)$$

where the left - handed side was the equality in equation (2) multiplied by a constant Z equating both sides. After cancelling out common terms on both sides of equation (7), Z becomes:

$$Z = [7.78 \times 10^{14}\ \mu m \cdot cm^2 \cdot K^5 / Watts] * [H_{\lambda peak} / T^5] \qquad (8)$$

Since and T remain constant with respect to wavelength, Z was carried outside the integral. Equation (4) became:

$$SR = Z \int_0^\infty [C_1 / \lambda^5 \cdot (e^x - 1)]\ d\lambda \qquad (9)$$

Finally, the spectral irradiance over all wavelengths is:

$$SR = Z \sigma T^4 \qquad (10)$$

For final evaluations, a short program, using the subroutine IRRADIANCE, was written (Appendix G).

Each of the former programs were written and run on the Orion MicroVAX III.

Resulting from S.A.S.S.E. are over 5 new data files which are necessary for specific sensors and/or further astrophysical studies for star sensors.

It was found that there are 17,056 certain stars (by defined parameters) that can be determined from the SAO and AFGL maps.

The brightest 500 stars by visual magnitude have been addressed and recorded alphabetically.

At the close of the summer of 1990 High School Apprenticeship Program, two key projects which compose the S.A.S.S.E. effort were achieved. Both the Alpha/Beta programs (which address and alphabetize the brightest stars) and D.U.M.B.O. (which matches the SAO and AFGL stars and computes spectral irradiance in specific wavebands) were successfully completed.

Further studies showed that the most effective star sensor would be one between 10E-15 and 10E-16 microns, allowing enough stars to be seen for positional sightings. Also, the purpose for the study has been altered due to the fact that the star sensor is now strictly passive; therefore, pattern matching is no longer plausible.

Lastly, two short studies were conducted to evaluate the number of stars which can be viewed under given circumstances. The first calculates the number of stars which can be viewed in different wavebands (Appendix I). The second evaluates the number of stars which may be viewed by a given sensor with different spectral N.E.F.D. values (Appendix J). In other words, it calculates the number of stars which are bright enough to see in a certain waveband.

The High School Apprenticeship Program has helped me to develop many diverse skills and to learn much about engineering and scientific fields. First of all, compiled with the knowledge of last summer, I became highly proficient in FORTRAN programming. Also, I learned many tasks of an astronomer, including reading star maps and identifying stars by address.

In addition to the astronomy, I had the opportunity to learn many astrophysical equation manipulations along with the theories supporting them. I learned how to convert right ascension and declination to practicable forms. Also, I gained a new understanding of the different types of magnitude and how to convert one to another. Next, I learned to recognize and use equations which are utile for spectral irradiance manipulations and calculations.

Finally, what I most appreciated about this summer opportunity is what one can not learn from books or classes, and that's the experiences and wisdom which was imparted to me from my mentor, Tim Poth, and others that I met. I talked to engineers (Aero's and Astro's) along with physicists (contractors and government) and from this learned of the differences which distinguish one from another. Additionally, I found out about different colleges, different places and companies to work at, and the "do's" and "don't's" of the college and professional fields. Conclusively, I am more confident in my choices for college, career, and future, reconfirming, beyond a shadow of a doubt, that I do not want to be and engineer but a physicist.

```
C
C
      PROGRAM MAXMAG 500
C
C ^^^^^^THIS PROGRAM READS IN THE REDUCED SAO MAP AND OUTPUTS THE
C ^^^^^^TOP 500 STARS BY VISUAL MAG
C
      DIMENSION DATA(600,15)
      OPEN(10,FILE='SAOMXNH.MAG',STATUS='OLD')
      OPEN(20,FILE='FIVC.DAT',STATUS='NEW')
      WRITE(20,50)
C
C ^^^^^^THIS READS IN THE 500 STARS ALLOWING FOR THE 36 DOUPLICATES
C ^^^^^^AND WRITES THEM TO A FILE^^^
C
      DO 10 I=1,536
         READ (10,100)(DATA(I,J),J=1,15)
         WRITE (20,200)I,(DATA(I,J),J=2,15)
 10   CONTINUE
 50   FORMAT(2X,'RANK',1X,' NUM',' L',1X,' RTASC',1X,' SND',2X,
     +       ' DECL',3X,' PM',2X,'VM','  ST',2X,'DV VMA'
     +       ' PMA CI1 CI2',5X,' RA',10X,'DEC')
 100  FORMAT(A5,A6,A1,F10.5,1X,A1,F10.5,1X,F4.1,1X,F4.1,1X,A3,
     +       2X,A1,2X,A1,3X,A1,5X,A3,3X,F10.8,1X,F11.8)
 200  FORMAT(I5,A6,A1,F10.5,1X,A1,F10.5,1X,F4.1,1X,F4.1,1X,A3,
     +       2X,A1,2X,A1,3X,A1,5X,A3,3X,F10.8,1X,F11.8)
      STOP
      END
```

```
      PROGRAM ALPHA
C
C  THIS PROGRAM PLACES THE TOP 500 STARS IN ALPHABETICAL ORDER BY
C  CONSTELATION AND DECENDING VISUAL MAGNITUDE
C
C  THIS PROGPAM ALLOWS THE USER TO INPUT UP TO EIGHTY CONSTELLATIONS
C  IN ANY ORDER AND CAUSES THE STARS AND THEIR DATA TO BE PLACED
C  IN THIS ORDER
C
C  JUNE 5,1990
C
C  BY NICKI COOK
C
      REAL RTASC,DECL,VM
      INTEGER N,RANK
      CHARACTER CONST*18,CONSTEL*18,STAR*15,SND*1,SYM*1,NUM*6
C
      OPEN (UNIT=4,FILE='ALPHA.DAT',STATUS='OLD')
      OPEN (UNIT=6,FILE='CONSTL.ABC',STATUS='OLD')
      OPEN (UNIT=8,FILE='BETA2.DAT',STATUS='NEW')
C
C****INPUTTING UP TO 80 CONSTELLATION NAMES****
C
      DO 99, M=1,80
        WRITE(*,*)'CONSTELATION NAME?'
        READ(*,10)CONST
 10     FORMAT(A18)
          DO 88, N=1,500
C
C****ALLOWS STARS FROM THE FILE OF TOP 200 STARS TO BE READ****
C
           IF(N.LT.201)THEN
           READ(6,20)RANK,NUM,SYM,CONSTEL,STAR,VM,RTASC,SND,DECL
           ENDIF
C
 20      FORMAT(I5,A6,2X,A1,2X,A18,1X,A15,F4.1,F10.5,1X,A1,F10.5)
C
C****ALLOWS STARS FROM THE FILE OF THE TOP STARS BY VM (300-500)
C****TO BE READ***
C
           IF(N.GT.200)THEN
           READ(4,20)RANK,NUM,SYM,CONSTEL,STAR,VM,RTASC,SND,DECL
           ENDIF
C
C****THIS CONDITIONAL MATCHES THE CONSTELLATION INPUTTED WITH
C****THE STARS IN THE DATA FILES FALLING IN THAT CONSTELLATION****
C
           IF(CONSTEL.EQ.CONST)THEN
C*****************************************************************
C
C****WRITES PERTINENT INFORMATION TO A FILE****
C
           WRITE(8,30) SYM,CONST,STAR,VM,RTASC,SND,DECL,RANK,NUM
 30      FORMAT(2X,A1,2X,A18,1X,A15,F4.1,F10.5,1X,A1,F10.5,2X,I5,A6)
           ENDIF
 88      CONTINUE
         REWIND(4)
         REWIND(6)
 99    CONTINUE
       CLOSE (4)
```

```
CLOSE (8)
STOP
END
```

```
      PROGRAM BETA

C  THIS PROGRAM PLACES THE 500 BRIGHTEST STARS IN THEIR APPROPRIATE
C  CONSTELLATION AND NAMES THE STAR WHILE GIVING VISUAL MAGNITUDES,
C  RIGHT ASCENSIONS, AND DECLINATION.
C
C  MAY 31, 1990
C
C  BY NICKI COOK
C
      REAL PM,VM,RTASC,DECL,RA,DC
      INTEGER RANK
      CHARACTER PMA,DV,VMA,L
      CHARACTER CONST*18,STAR*15,SND*1,ST*3,CI12*3,SYM*1,CHK*3,NUM*6
C
      OPEN (UNIT=2, FILE='SAOMXNH.MAG', STATUS='OLD')
      OPEN (UNIT=4, FILE='ALPHAZ.DAT', STATUS='NEW')

C
C  Writing headings to file**************************************

C
      WRITE (4,*) 'RANK  NUM   SYM  CONSTELATION       STAR
     + VM  RTASC     DEC'

C
C  This reads pertinent info from map and matches it with star and
C  constelation info.*******************************************
C

      DO 99,N=1,501
         READ(2,10)RANK,NUM,L,RTASC,SND,DECL,PM,VM,ST,DV,VMA,PMA,CI12,RA
 10      FORMAT(I5,A6,A1,F10.5,1X,A1,F10.5,1X,F4.1,1X,F4.1,1X,A3,2X,A1,2X,
     +       A1,3X,A1,5X,A3,3X,F10.8,1X,F11.8)
         IF(RANK.LT.236)GO TO 99
         WRITE(*,22)RANK
 22      FORMAT(' PLEASE TYPE CONSTELLATION NAME FOR STAR # ',I5,
     +       '(0 FOR NONE)')
         READ(*,'(A)')CONST
         CALL CHECK (CONST)
         WRITE(*,*)'SYMBOL (0 FOR NONE)'
         READ(*,'(A)')SYM
         CALL CHECK (SYM)
         WRITE(*,*)'STAR NAME (0 FOR NONE)'
         READ(*,'(A)')STAR
         CALL CHECK (STAR)


C   Writing star info to file*********************************


         WRITE(4,40)N,NUM,SYM,CONST,STAR,VM,RTASC,DECL
 40      FORMAT(I5,A6,2X,A1,2X,A18,1X,A15,F4.1,F10.5,1X,F10.5)


 99   CONTINUE
C
       CLOSE(2)
       CLOSE(4)
C
```

```
      END
C     Checks for 0's in order to print blanks in file**********
      SUBROUTINE CHECK (CHK)
        IF (CHK.EQ.'0') THEN CHK='---'
      RETURN
      END
```

RANK	NUM	SYM	CONSTELLATION	STARNAME	VM	RT ASCEN	DECLIN
1	1881	A	CANIS MAJOR	SIRIUS	-1.6	100.73631	- 16.64621
2	4480	A	CANOPUS	------	-0.9	95.71039	- 52.66763
3	2838	A	CENTAURUS	-----	0.1	219.04689	- 60.63023
4	7174	A	LYRA	VEGA	0.1	278.81107	38.73602
5	0186	A	AURIGA	CAPELLA	0.2	78.24777	45.94946
6	0944	A	BOOTES	ARCTURUS	0.2	213.34479	19.44193
7	1907	B	ORION	RIGEL	0.3	78.03329	- 8.25796
8	5756	A	CANIS MINOR	PROCYON	0.5	114.17132	5.35467
9	2481	A	ACHERNAR	-----	0.6	23.96378	- 57.49026
10	3271	A	ORION	BETELGEUSE	0.6	88.11587	7.39942
11	2582	B	CENTAURUS	-----	0.9	210.06879	- 60.13280
12	5122	A	AQUILA	ALTAIR	0.9	297.08578	8.73492
13	4415	A	SCORPIUS	ANTARES	1.1	246.58420	- 26.32277
14	4027	A	TAURUS	ALDERBARON	1.1	68.26207	16.41042
15	7923	A	VIRGO	SPICA	1.2	200.63875	- 0.90093
16	9666	B	GEMINI	POLLUX	1.2	115.56465	28.14864
17	9941	A	CYGNUS	DENEB	1.3	309.93143	45.10086
18	1524	A	PISCIS AUSTRINUS	FORMALHAUT	1.3	343.72305	- 29.88772
19	8967	A	LEO	REGULUS	1.3	151.42769	12.21237
20	0259	B	SOUTHERN CROSS	-----	1.5	191.19598	- 59.41570
21	1904	A	SOUTHERN CROSS	-----	1.6	185.95027	- 62.82206
22	0198	A	GEMINI	CASTOR	1.6	112.85272	31.99974
23	0019	C	CRUX	-----	1.6	187.09470	- 56.83342
24	2676	E	CANIS MAJOR	-----	1.6	104.16495	- 28.90284
25	8553	E	URSA MAJOR	ALIOTH	1.7	192.95868	56.23087
26	8954	C	SCORPIUS	-----	1.7	262.55261	- 37.06934
27	5932	E	CARINA	-----	1.7	125.37243	- 59.34805
28	2740	C	ORION	BELLATRIX	1.7	80.61179	6.30603
29	0495	B	CARINA	-----	1.8	138.16525	- 69.51101
30	2346	D	ORION	MINTAKA	1.8	83.41865	- 1.23231
31	7168	B	TAURUS	NATH	1.8	80.78213	28.56715
32	8787	A	PERSEUS	MIRFAK	1.9	50.18517	49.68501
33	4752	N	URSA MAJOR	BENETNASCH	1.9	206.39290	49.56226
34	5912	C	GEMINI	ALHENA	1.9	98.70582	16.44372
35	3700	A	TRIANGULUMAUSTRALE	-----	1.9	250.83775	- 68.93886
36	9504	C	VELA	-----	1.9	121.99777	- 47.18842
37	1428	B	CANIS MAJOR	-----	2.0	95.12412	- 17.92978
38	3047	D	CANIS MAJOR	-----	2.0	106.58932	- 26.31260
39	2444	E	ORION	ALNILAM	2.0	84.55851	- 1.96749
40	4129	E	CORONA BOREALIS	-----	2.0	239.35210	26.06084
41	8201	T	SCORPIUS	-----	2.0	263.43073	- 42.96810
42	0091	E	SAGITTARIUS	-----	2.0	275.21329	- 34.41026
43	9825	O	CETUS	MIRA	2.0	34.20435	- 3.20372
44	6232	D	VELA	-----	2.0	130.83076	- 54.52452
45	5384	A	URSA MAJOR	DUBHE	2.0	165.16487	62.02133
46	6574	A	PAVO	-----	2.1	305.42621	- 56.89723
47	308	A	URSA MINOR	POLARIS	2.1	27.20328	89.02882
48	7448	S	SAGITTARIUS	-----	2.1	283.04135	- 26.36064
49	2932	A	OPHIUCHUS	RAS ALHAGUE	2.1	263.15289	12.59498
50	0750	B	AURIGA	COLURE	2.1	88.96491	44.94464
51	3765	A	ANDROMEDA	SIRRAH***	2.1	1.44934	28.81448
52	0878	C	VELA	-----	2.2	136.53879	- 43.22989
53	6871	A	HYDRA	ALPHARD	2.2	141.28246	- 8.44096
54	2542	K	ORION	SAIPH	2.2	86.34589	- 9.68596
55	9809	B	LEO	DENEBOLA	2.2	176.62752	14.85161
56	8102	B	URSA MINOR	KOCHAB	2.2	222.70685	74.35988
57	0992	A	GRUS	-----	2.2	331.27286	- 47.20404
58	5151	A	ARIES	HAMAL	2.2	31.08715	23.22696

59	7420	B	CETUS	DENEBKAITOS	2.2	10.27005	-	18.26077
60	1258	B	GRUS	-----	2.2	339.92267	-	47.14672
61	9528	C	CYGNUS	-----	2.3	305.10803		40.09571
62	5188	T	CENTAURUS	-----	2.3	210.93282	-	36.12490
63	3893	A	CORONA BOREALIS	GEMMA	2.3	233.14227		26.88188
64	7734	C	ANDROMEDA	ALMAK	2.3	30.20490		42.09084
65	6808	1	CARINA	-----	2.3	136.93796	-	59.06491
66	8752	Z	PUPPIS	-----	2.3	120.45647	-	39.86134
67	3603	C	CENTAURUS	-----	2.4	189.68707	-	48.68537
68	0653	C	DRACO	ETAMIN	2.4	268.86075		51.49404
69	1133	B	CASSEOPEIA	-----	2.4	1.62390		58.87410
70	8737	Z	URSA MAJOR	MIZAR	2.4	200.47882		55.18596
71	3651	N	CANIS MAJOR	-----	2.4	110.52905	-	29.20446
72	8078	E	SCORPIUS	-----	2.4	251.73003	-	34.20432
73	4471	B	ANDROMEDA	MIRACH	2.4	16.73107		55.35604
74	5093	A	PHOENIX	-----	2.4	5.95429	-	42.57737
75	7876	B	URSA MAJOR	MERAK	2.4	164.70938		56.65087
76	7029	E	PEGASUS	ENIT	2.5	325.43231		9.64491
77	9163	K	SCORPIUS	-----	2.5	264.75644	-	39.00632
78	1609	A	CASSEOPEIA	SCHEDIR	2.5	9.41380		56.26350
79	8179	C	URSA MAJOR	PHEODA	2.5	177.80235		53.97277
80	2220	D	ORION	-----	2.5	82.36257	-	0.33456
81	4014	D	SCORPIUS	-----	2.5	239.34299	-	22.48094
82	1047	E	CENTAURUS	-----	2.6	204.17635	-	53.21287
83	1298	C	LEO	ALGEIBA	2.6	154.30467		20.09524
84	0474	E	CYGNUS	-----	2.6	311.04654		33.78199
85	9302	A	CEPHEUS	ALDERAMIN	2.6	319.34659		62.37326
86	1727	D	LEO	ZOSMA	2.6	167.86285	-	20.79796
87	5044	N	LUPUS	-----	2.6	218.08044	-	41.93938
88	8378	A	PEGASUS	MARKAB	2.6	345.56693		14.93585
89	0332	N	OPHIUCHUS	-----	2.6	256.87692	-	15.66469
90	6891	H	VELA	-----	2.6	140.14105	-	54.79647
91	0981	B	PEGASUS	SCHEAT	2.6	345.33655		27.81124
92	3500	E	BOOTES	IZAR	2.7	220.70059		27.28407
93	0430	B	LIBRA	-----	2.7	228.57805	-	9.19970
94	0006	Z	OPHIUCHUS	-----	2.7	238.60057	-	10.46745
95	0547	A	LEPUS	ARNEB	2.7	82.63081	-	17.85672
96	8636	T	AURIGA	-----	2.7	89.07761		37.21112
97	7795	P	PUPPIS	-----	2.7	108.84396	-	37.00664
98	7600	-	SAGITTARIUS	-----	2.7	284.85788	-	29.95355
99	5012	B	ARIES	CHERATAN	2.7	27.96808		20.56447
100	4735	U	PUPPIS	-----	2.8	102.17368	-	50.55430
101	6059	A	COLUMBA	-----	2.8	84.45921	-	34.09955
102	0915	B	CORVUS	-----	2.8	187.93892	-	23.12045
103	4725	B	ARA	-----	2.8	260.28442	-	55.48503
104	2268	D	CASSEOPEIA	-----	2.8	20.63124		59.97621
105	1482	C	CASSEOPEIA	-----	2.8	13.41802		60.44650
106	2321	M	VELA	-----	2.8	161.15341	-	49.15546
107	0920	A	CETUS	MENKAB	2.8	44.91560		3.89476
108	5223	C	AQUILA	TARAZED	2.8	295.97046		10.49011
109	5235	B	LUPUS	-----	2.8	223.81161	-	42.93391
110	0 56	N	BOOTES	-----	2.8	208.07570		18.64760
111	66 1	D	SAGITTARIUS	-----	2.8	274.44815	-	29.85130
112	442	B	HERCULES	-----	2.8	247.01711		21.59727
113	115	A	SERPENS CAPUT	UNULA	2.8	235.45062		6.58165
114	7176	C	CORVUS	-----	2.8	183.30768	-	17.26440
115	8896	L	SCORPIUS	-----	2.8	261.84027	-	37.25793
116	2671	B	OPHIUCHUS	-----	2.9	265.25018		4.58662
117	1974	A	MASCA	-----	2.9	188.54456	-	68.86023
118	5193	A	TUCANA	-----	2.9	333.77371	-	60.50973

119	8840	A	LIBRA	-----	2.9	222.02686	-	15.83516
120	9682	B	SCORPIUS	AKRAB	2.9	240.63129	-	19.67012
121	6841	L	SAGITTARIUS	-----	2.9	276.22107	-	25.45116
122	4481	T	SCORPIUS	-----	2.9	248.19127	-	28.11404
123	8592	B	PERSEUS	ALGOL	2.9	46.22648		40.76457
124	5217	P	PUPPIS	-----	2.9	121.35342	-	24.15897
125	7522	I	AURIGA	-----	2.9	73.43326		33.08887
126	5670	B	HYDRUS	-----	2.9	5.78906	-	77.53569
127	3257	A	CANES VENATICI	COR CAROLI	2.9	193.42282		38.58801
128	6799	Z	PERSEUS	-----	2.9	57.74566		31.73682
129	7074	N	DRACO	-----	2.9	245.82698		61.62697
130	1794	B	ERIDANUS	-----	2.9	76.34735	-	5.14959
131	5128	A	LUPUS	-----	2.9	219.64804	-	47.17481
132	2323	I	ORION	-----	2.9	83.24636	-	5.94118
133	4371	I	CENTAURUS	-----	2.9	199.44452	-	36.44913
134	1781	C	PEGASUS	-----	2.9	2.66436		14.90571
135	8917	C	VIRGO	-----	2.9	189.78105	-	1.17544
136	9689	D	CENTAURUS	-----	2.9	181.43936	-	50.44398
137	4203	C	BOOTES	-----	3.0	217.51599		38.52615
138	4461	Z	AQUILA	-----	3.0	285.77771		13.78774
139	7756	P	SAGITTARIUS	-----	3.0	286.69772	-	21.10486
140	8796	D	CYGNUS	-----	3.0	295.85278		45.00784
141	7067	H	VELA	-----	3.0	142.42529	-	56.81322
142	5485	Z	HERCULES	-----	3.0	249.84995		31.69226
143	5938	Z	LUPUS	-----	3.0	232.94995	-	41.00027
144	0429	B	DRACO	-----	3.0	262.32504		52.33768
145	3346	B	TRIANGULUMAUSTRALE	-----	3.0	237.67908	-	63.27850
146	0457	B	LEPUS	NIHAL	3.0	81.52538	-	20.79803
147	8474	A	HYDRUS	-----	3.0	29.29876	-	61.81262
148	1083	T	CARINA	-----	3.0	160.29189	-	64.13205
149	8069	A	ARA	-----	3.0	261.99313	-	49.83878
150	7336	Z	TAURUS	-----	3.0	83.66360		21.11389
151	4644	D	CAPRICORNUS	-----	3.0	326.07080	-	16.35512
152	0384	E	VIRGO	VINDEMIATRIX	3.0	194.92171		11.22747
153	3987	F	SCORPIUS	-----	3.0	238.95541	-	25.97171
154	6199	N	PLEIADS	ALCYONE	3.0	56.12677		23.95210
155	6840	E	PERSEUS	-----	3.0	58.62263		39.86737
156	1052	D	OPHIUCHUS	-----	3.0	242.93053	-	3.56706
157	4336	S	SCORPIUS	-----	3.1	244.53612	-	25.47448
158	0734	N	ANDROMEDA	-----	3.1	340.16360		29.95928
159	9696	C	SAGITTARIUS	-----	3.1	270.64865	-	30.42661
160	4538	Z	CENTAURUS	-----	3.1	208.10213	-	47.04302
161	3629	W	URSA MAJOR	-----	3.1	166.71494		44.77017
162	3097	C	TRIANGULUMAUSTRALE	-----	3.1	228.55244	-	68.49689
163	0695	N	CARINA	-----	3.1	146.46320	-	64.83950
164	8220	C	URSA MINOR	-----	3.1	230.19728		72.01193
165	2839	S	CANIS MAJOR	-----	3.1	105.23392	-	23.75902
166	5457	B	AQUARIUS	-----	3.1	322.23187	-	5.79213
167	8549	S	PUPPIS	-----	3.1	108.00286	-	44.55733
168	2630	I	URSA MAJOR	-----	3.1	133.94844		48.23946
169	5306	B	TRIANGULUM	-----	3.1	31.63991		34.75181
170	3789	C	PERSEUS	-----	3.1	45.28998		53.31230
171	1004	E	LEO	-----	3.1	145.75415		24.00545
172	9053	D	PERSEUS	-----	3.1	54.83850		47.62954
173	8420	I	SCORPIUS	-----	3.1	266.02124	-	40.10971
174	7323	D	CORVUS	ALGORAB	3.1	186.81825	-	16.23726
175	6698	Z	CANIS MAJOR	-----	3.1	94.59811	-	30.03996
176	4315	Z	ARA	-----	3.1	253.61842	-	55.91351
177	5456	B	CANIS MINOR	-----	3.1	111.10982		8.39164
178	9791	E	SOUTHERN CROSS	-----	3.1	183.11917	-	58.47078

179	8102	M	SCORPIUS	-----	3.1	252.11949	-	37.96359
180	8222	D	DRACO	-----	3.2	288.13678		67.57362
181	4951	D	HERCULES	-----	3.2	258.24399		24.89688
182	3374	C	GRUS	-----	3.2	327.72687	-	37.60099
183	6029	C	HYDRUS	-----	3.2	56.99777	-	74.39265
184	3310	M	URSA MAJOR	-----	3.2	154.83945		41.75174
185	0300	A	INDUS	-----	3.2	308.51471	-	47.46748
186	0531	E	CORVUS	-----	3.2	181.88716	-	22.34170
187	6240	B	COLUMBA	-----	3.2	87.29887	-	35.78604
188	8682	E	GEMINI	MEBSUTA	3.2	100.21401		25.18248
189	5862	A	AQUARIUS	-----	3.2	330.80396	-	0.56350
190	8071	V	PUPPIS	-----	3.2	99.05748	-	43.15114
191	7301	B	CYGNUS	ALBIREO	3.2	292.17593		27.85346
192	9283	C	ERIDANUS	-----	3.2	58.92363	-	13.64949
193	8297	M	GEMINI	-----	3.2	94.98373		22.54111
194	7365	Z	DRACO	-----	3.2	257.15894		65.77612
195	9957	N	SAGITTARIUS	-----	3.2	273.56073	-	36.77875
196	7289	T	URSA MAJOR	OWL NEBULA	3.3	142.38121		51.90644
197	9318	G	SCORPIUS	-----	3.3	266.61322	-	37.02934
198	0057	B	CEPHEUS	-----	3.3	322.00552		70.34107
199	1414	A	LYNX	-----	3.3	139.50360		34.60516
200	0237	M	LEPUS	-----	3.3	77.67089	-	16.26332

RANK	NUM	SYM	CONSTELLATION	STAR NAME	VM	RT ASCEN	DECLIN
201	7239	R	SAGITTARUS	0	3.3	280.63318	27.04402
202	1543	C	HYDRA	0	3.3	199.04945	22.90830
203	8755	S	PUPPIS	0	3.3	111.91090	43.19929
204	9647	A	PICTOR	0	3.3	101.91969	61.88733
205	7663	C	LYRA	0	3.3	284.26788	32.61979
206	3481	B	CAPRICORNUS	0	3.3	304.55084	14.94071
207	2106	P	ORION	0	3.3	71.78079	6.87563
208	4471	M	CENTAURUS	0	3.3	206.64857	42.22546
209	2019	B	MUSCA	0	3.3	190.79788	67.83477
210	0026	V	AURIGA	0	3.3	75.75089	41.16899
211	7264	Z	HYDRA	0	3.3	133.18788	6.13701
212	6256	V	HYDRA	0	3.3	161.78888	15.93151
213	1086	E	OPHIUCHUS	0	3.3	243.91812	4.57210
214	1472	L	CENTAURUS	0	3.3	173.36595	62.74294
215	0051	E	LEPUS	0	3.3	75.83568	22.43690
216	6074	T	CANIS MAJOR	0	3.4	100.62069	12.95120
217	0905	Q	CARINA	0	3.4	153.85242	61.08197
218	5890	P	HERCULES	0	3.4	258.32590	36.86440
219	5344	B	LUPUS	0	3.4	223.97462	41.90491
220	2031	E	CASSIOPEIA	0	3.4	27.69324	63.42497
221	9512	D	CRATER	0	3.4	167.90443	15.70316
222	2853	A	CIRCINUS	0	3.4	219.60991	64.75900
223	2241	V	SERPENS CAUDA	0	3.4	274.68027	2.91339
224	5691	E	LUPUS	0	3.4	229.52022	40.46803
225	5320	T	OPHIUCHUS	0	3.4	259.73398	24.95144
226	5365	B	PHOENIX	0	3.4	15.96399	46.98605
227	1070	Z	CYGNUS	0	3.4	317.70148	30.02098
228	2071	N	ORION	0	3.4	80.49032	2.44155
229	4150	T	AQUILA	0	3.4	302.18121	0.97113
230	3139	S	LIBRA	0	3.4	225.28436	25.08674
231	9420	Z	VIRGO	0	3.4	203.03564	0.34098
232	7683	U	SAGITTARUS	0	3.4	285.95483	27.74532
233	5516	C	PHOENIX	0	3.4	21.54907	43.57387
234	8969	A	RECTICULUM	0	3.4	63.44393	62.59862
235	7707	N	SCORPIUS	0	3.4	257.14172	43.17531
236	1962	H	OPHIUCHUS	0	3.4	253.82483	9.45134
237	0818	C	CEPHEUS	0	3.4	354.31888	77.35326
238	4603	D	AQUILA	0	3.4	290.74414	3.01361
239	8315	D	URSA MAJOR	0	3.4	183.23993	57.31026
240	6113	T	ERIDANUS	0	3.4	44.09118	40.50421
241	5375	L	AQUARIUS	0	3.5	342.99960	16.08720
242	4469	M	CENTAURUS	0	3.5	206.62390	41.43933
243	3268	L	URSA MAJOR	0	3.5	153.52231	43.16488
244	9294	D	GEMINI	0	3.5	109.28444	22.07613
245	5397	M	HERCULES	0	3.5	266.12500	27.74872
246	2680	N	OPHIUCHUS	0	3.5	258.09146	14.44593
247	3021	L	AQUILA	0	3.5	285.89871	4.95909
248	9955	E	AURIGA	0	3.5	74.59387	43.75149
249	4058	D	ANDROMEDA	0	3.5	9.16191	30.58772
250	2004	V	SERPENS CAUDA	0	3.5	269.06805	9.76919
251	4573	O	URSA MAJOR	0	3.5	126.53180	60.88738
252	2304	A	LUPUS	0	3.5	227.16983	51.91050
253	1695	S	LIBRA	0	3.5	201.74367	23.02348
254	4726	C	HYDRA	0	3.5	260.29471	56.33301
255	4589	D	BOOTES	0	3.5	228.37132	33.50033
256	9520	I	DRACO	0	3.5	230.95309	59.14060
257	7112	E	HYDRA	0	3.5	131.03232	6.60345
258	2244	P	HYDRA	0	3.5	210.87935	26.44236

259	4601	F	PUPPIS	0	3.5	116.79758	24.73312
260	3564	A	DORADO	0	3.5	68.22832	55.14772
261	4996	A	TRIANGULUM	0	3.6	27.55600	29.33615
262	7986	U	CETUS	0	3.6	25.43608	16.20014
263	8103	Z	PEGASUS	0	3.6	339.74164	10.56983
264	5504	N	HERCULES	0	3.6	250.29491	39.01633
265	9570	T	GEMINI	0	3.6	102.37363	34.02349
266	1265	Z	LEO	0	3.6	153.47816	23.66721
267	0885	J	CARINA	0	3.6	153.13770	69.78926
268	7504	F	SAGITTARIUS	0	3.6	283.68680	21.17413
269	6273	A	DRACO	0	3.6	210.75818	64.61433
270	6552	R	LUPUS	0	3.6	229.65630	36.08144
271	1006	G	CARINA	0	3.6	157.56036	61.42769
272	4733	D	PAVO	0	3.6	300.96021	66.31207
273	4902	U	ERIDANUS	0	3.6	64.00015	33.91935
274	0787	M	OPHIUCHUS	0	3.6	236.75186	3.27860
275	8955	N	LEO	0	3.6	151.15222	17.00727
276	3954	E	TAURUS	0	3.6	66.42327	19.07121
277	8116	M	SCORPIUS	0	3.6	252.23582	37.93422
278	4020	N	PAVO	0	3.6	265.20517	64.70274
279	1862	E	SOUTHERN CROSS	0	3.6	184.66174	60.12502
280	3957	T	TAURUS	0	3.6	66.45092	15.76161
281	0228	B	POLARIS VOLAN	0	3.6	126.29981	65.96972
282	7000	D	MUSCA	0	3.6	194.69989	71.27978
283	7632	N	CETUS	0	3.6	16.51842	10.44683
284	5337	B	BOOTES	0	3.6	225.01527	40.58689
285	0707	C	CETUS	0	3.6	40.17665	3.02612
286	0700	F	SERPENS CAUDA	0	3.6	263.68030	15.3688(
287	2609	O	ANDROMEDA	0	3.6	344.90363	42.05698
288	5635	H	CARINA	0	3.6	118.87678	52.84743
289	4137	Z	CEPHEUS	0	3.6	332.27890	57.95431
290	4862	B	PAVO	0	3.6	310.11942	66.38483
291	1234	W	VELA	0	3.6	142.18205	40.24702
292	9019	N	CEPHEUS	0	3.6	311.06870	61.64413
293	1732	N	CASSIOPEIA	0	3.6	11.51519	57.55087
294	6693	I	VELA	0	3.6	137.41263	58.76159
295	6746	C	GEMINI	0	3.6	108.80499	16.63226
296	7208	N	NORMA	0	3.6	239.20010	38.25555
297	2484	N	PISCES	0	3.7	22.20082	15.08872
298	5712	E	LUPUS	0	3.7	229.81880	44.51117
299	2797	S	CANIS MAJOR	0	3.7	104.93159	27.86203
300	7593	E	GRUS	0	3.7	341.38629	51.58035
301	6316	B	DELPHINUS	0	3.7	308.80087	14.42009
302	7340	T	PEGASUS	0	3.7	331.91916	5.95115
303	0801	Z	LEPUS	0	3.7	86.17207	14.83922
304	3142	0	OPHIUCHUS	0	3.7	271.24438	9.55526
305	8135	N	GEMINI	0	3.7	92.96461	22.52316
306	1566	Z	CASSIOPEIA	0	3.7	8.54305	53.62206
307	2558	F	HYDRA	0	3.7	172.63443	31.58074
308	0803	0	VELA	0	3.7	135.60709	46.89792
309	0686	D	ERIDANUS	0	3.7	55.21265	9.93140
310	0816	B	PEGASUS	SCHEAT	3.7	341.89664	24.33714
311	2573	H	ERIDANUS	0	3.7	28.50343	51.85715
312	6138	Y	PERSEUS	0	3.7	45.49081	38.64802
313	0268	I	CEPHEUS	0	3.7	341.97336	65.93716
314	9653	H	GEMINI	0	3.7	115.35773	24 51958
315	6784	B	INDUS	0	3.7	312.72949	58.64450
316	2486	V	URSA MAJOR	0	3.7	168.94562	33.36737
317	1952	U	ORION	0	3.7	78.79417	6.89696
318	9546	A	PYXIS	0	3.7	130.39537	33.00518

319	3831	B	CORONA BOREALIS	0	3.7	231.44144	29.27702	
320	7522	R	VELA	0	3.7	148.77603	54.32913	
321	2661	H	URSA MAJOR	0	3.7	135.05528	47.35577	
322	5596	0	0	0	3.7	41.75848	27.05571	
323	9087	H	DRACO	0	3.7	275.48950	72.71171	
324	6164	O	VELA	0	3.7	129.71515	52.74364	
325	1725	B	SERPENS CAPUT	0	3.7	235.96945	15.57706	
326	2754	C	OPHIUCHUS	0	3.7	266.34592	2.72452	
327	9674	D	VIRGO	0	3.7	193.27078	3.66878	
328	8398	0	0	0	3.7	115.86803	37.84625	
329	4168	N	ARA	0	3.7	251.36418	58.95455	
330	5500	C	SAGITTA	0	3.7	299.13309	19.35513	
331	7948	V	OCTANS	0	3.7	323.99945	77.61395	
332	2921	R	ORION	0	3.7	83.09547	9.90232	
333	8877	B	RECTICULUM	0	3.8	55.89151	64.96394	
334	4810	E	AQUARIUS	0	3.8	311.24258	9.68006	
335	3649	0	0	0	3.8	233.89471	29.61488	
336	1683	0	0	0	3.8	346.69574	21.44415	
337	6872	I	HERCULES	0	3.8	264.51276	46.03202	
338	9274	T	CETUS	0	3.8	20.38068	8.44088	
339	6710	A	CAELUM	0	3.8	63.08580	42.41676	
340	1575	L	MUSCA	0	3.8	175.80843	66.45140	
341	8694	I	CETUS	0	3.8	4.21993	9.10096	
342	3427	A	CAPRICORNUS	0	3.8	303.82031	12.70125	
343	0648	0	0	0	3.8	220.92952	2.10249	
344	0759	D	LEPUS	0	3.8	85.59439	22.46342	
345	9082	A	PUPPIS	0	3.8	117.62428	40.44591	
346	5590	F	HERCULES	0	3.8	268.95505	29.25204	
347	9311	B	DORADO	0	3.8	83.29721	62.52228	
348	2107	0	0	0	3.8	244.92807	19.26925	
349	1195	E	ERIDANUS	0	3.8	51.11387	9.55977	
350	7402	Z	SCORPIUS	0	3.8	252.76477	42.27774	
351	6228	0	0	0	3.8	56.54596	23.90213	
352	4908	0	0	0	3.8	141.90236	63.28197	
353	4202	0	0	0	3.8	217.41852	30.59004	
354	7527	T	HYDRA	0	3.8	137.94090	2.52629	
355	9023	A	TELESCOPIUM	0	3.8	275.81656	45.99809	
356	3619	V	LUPUS	0	3.8	233.49568	27.97094	
357	1121	F	MICROSCOPIUM	0	3.8	318.19815	37.83119	
358	5259	D	SAGITTA	0	3.8	296.28928	18.40964	
359	1218	E	SERPENS CAPUT	0	3.8	237.08018	4.62690	
360	2142	M	ERIDANUS	0	3.8	72.13509	5.52121	
361	5750	0	0	0	3.8	271.39758	28.75439	
362	6569	B	CANCER	0	3.8	123.45117	9.34104	
363	1299	C	LEO	0	3.8	154.30583	20.09436	
364	2696	R	ERIDANUS	0	3.8	33.68092	51.74303	
365	0957	N	LEPUS	0	3.8	88.53147	14.17546	
366	4560	C	CAPRICORNUS	0	3.8	324.33072	16.88918	
367	7258	H	CANIS MAJOR	0	3.8	101.99301	32.44963	
368	6362	L	AQUARIUS	0	3.8	342.50156	7.84615	
369	5785	L	HYDRA	0	3.8	152.03728	12.10626	
370	0564	E	ERIDANUS	0	3.8	52.64315	9.62632	
371	3282	0	0	0	3.8	108.19538	26.68472	
372	3945	D	ARA	0	3.8	261.64471	60.64470	
373	1391	0	LYNX	0	3.8	138.93477	37.01543	
374	1172	O	TAURUS	0	3.8	50.52967	8.85424	
375	7193	A	APUS	0	3.8	220.38747	78.83490	
376	2406	S	ORION	0	3.8	84.05853	2.62736	
377	6605	0	0	0	3.8	169.20955	14.50768	
378	9076	B	VIRGO	0	3.8	177.02245	2.04656	

379	8709	O	LEO	0	3.8	144.62086	10.12073
380	7375	0	0	0	3.8	23.72765	48.37574
381	6131	0	0	0	3.8	55.47523	23.95773
382	5092	0	0	0	3.9	5.93711	43.95713
383	5502	D	AURIGA	0	3.9	88.85216	54.28337
384	4281	M	ANDROMEDA	0	3.9	13.49233	38.22861
385	0424	F	CYGNUS	0	3.9	315.77747	43.72759
386	1658	L	SERPENS CAPUT	0	3.9	247.09726	2.09196
387	8355	Y	LEO	0	3.9	157.54485	9.56451
388	8574	0	0	0	3.9	162.86415	58.58729
389	4983	E	PHOENIX	0	3.9	1.71990	46.02325
390	6003	0	0	0	3.9	275.39142	21.74566
391	2484	V	URSA MAJOR	0	3.9	168.87984	31.81080
392	7593	H	DRACO	0	3.9	187.83980	70.06361
393	8085	C	PISCES	0	3.9	348.64304	3.00879
394	7407	C	APUS	0	3.9	246.42825	78.78897
395	0926	D	LEPUS	0	3.9	87.29247	20.88201
396	6357	A	DELPHINUS	0	3.9	309.32864	15.73453
397	7643	0	0	0	3.9	285.42184	21.81653
398	1579	O	ERIDANUS	0	3.9	60.12327	5.85208
399	1826	C	SERPENS CAPUT	0	3.9	238.53523	15.82355
400	5235	B	AQUILA	0	3.9	298.21411	6.28051
401	1702	I	CYGNUS	0	3.9	292.11115	51.62249
402	5716	E	HERCULES	0	3.9	254.59363	30.99887
403	7451	B	LYRA	0	3.9	282.05807	33.30347
404	9078	0	ERIDANUS	0	3.9	55.44700	42.42240
405	4542	A	LACERTA	0	3.9	337.30649	50.02499
406	3886	H	URSA MAJOR	0	3.9	175.85429	48.05671
407	0631	F	DRACO	0	3.9	268.16541	56.87991
408	8059	Z	CETUS	0	3.9	27.24770	10.58136
409	7401	V	URSA MAJOR	0	3.9	146.86293	59.27509
410	6374	C	POLARIS VOLAN	0	3.9	107.29501	70.41798
411	3868	C	TAURUS	0	3.9	64.23605	15.50851
412	1145	Z	BOOTES	0	3.9	219.68974	13.94180
413	9031	Z	GEMINI	0	3.9	105.28594	20.64540
414	0341	Z	CAPRICORNUS	0	3.9	320.95392	22.62898
415	3958	D	CORONA BOREALIS	0	3.9	235.16026	26.45288
416	6438	Z	POLARIS VOLAN	0	3.9	115.61243	72.48630
417	9374	I	GEMINI	0	3.9	110.65573	27.89921
418	5242	T	TELESCOPIUM	0	3.9	270.68430	50.09709
419	2297	ζ	LEO MINOR	0	3.9	162.63013	34.48489
420	2197	P	ORION	0	3.9	72.91109	2.36034
421	6028	U	HERCULES	0	3.9	244.55878	46.43156
422	6673	0	0	0	3.9	55.29416	32.13152
423	5148	V	ERIDANUS	0	3.9	68.40131	30.66360
424	9966	Z	AURIGA	0	3.9	74.74450	41.00494
425	4134	B	PICTOR	0	3.9	86.52473	51.08382
426	3897	D	TAURUS	0	3.9	65.01185	17.42689
427	5924	E	COLUMBA	0	3.9	82.35902	35.50611
428	0665	C	CETUS	0	4.0	39.22910	0.11389
429	6856	F	PERSEUS	0	4.0	58.92844	35.64903
430	6485	T	HERCULES	0	4.0	268.63409	37.25602
431	0274	0	0	0	4.0	313.82675	40.97385
432	6044	C	AQUARIUS	0	4.0	334.76840	1.63994
433	3719	L	TAURUS	0	4.0	59.47657	12.35059
434	5536	D	PHOENIX	0	4.0	22.29303	49.33200
435	8721	N	VIRGO	0	4.0	184.33650	0.38907
436	6735	H	COLUMBA	0	4.0	95.07116	33.41002
437	4545	0	0	0	4.0	202.03387	39.14977
438	9116	N	CYGNUS	0	4.0	298.60718	34.94951

439	8751	0	0	0
440	0374	0	0	0
441	3204	L	ANDROMEDA	0
442	2512	0	0	0
443	9809	D	POLARIS VOLAN	0
444	8738	0	0	0
445	4577	M	CENTAURUS	0
446	6230	E	DELPHINUS	0
447	3957	0	SCORPIUS	0
448	8513	J	PISCES	0
449	9781	0	0	0
450	8609	0	0	0
451	3955	C	TAURUS	0
452	3693	M	CEPHEUS	0
453	3560	0	0	0
454	5896	0	MONOCEROS	0
455	9336	0	0	0
456	0090	M	VIRGO	0
457	3013	0	OPHIUCHUS	0
458	5426	P	LUPUS	0
459	9587	I	LEO	0
460	9540	E	DRACO	0
461	0197	0	0	0
462	6497	M	SAGITTARIUS	0
463	7650	N	COLUMBA	0
464	8460	0	0	0
465	0614	0	0	0
466	9490	B	PYXIS	0
467	8373	A	FORNAX	0
468	1537	0	0	0
469	0238	I	PEGASUS	0
470	6955	C	MUSCA	0
471	3655	N	PERSEUS	0
472	9337	0	0	0
473	6155	0	PLEIADS	0
474	8813	0	CENTAURUS	0
475	5159	N	AQUILA	0
476	4508	D	CEPHEUS	0
477	1154	D	GRUS	0
478	3474	D	CIRCINUS	0
479	9370	S	LEO	0
480	8804	C	LIBRA	0
481	0990	B	CORONA AUSTRALIS	0
482	0422	0	VELA	0
483	5871	0	0	0
484	2303	C	CANIS MAJOR	0
485	0775	L	PEGASUS	0
486	5980	0	0	0
487	1895	N	VELA	0
488	6181	N	AQUARIUS	0
489	5427	C	TRIANGULUM	0
490	0204	0	0	0
491	2199	M	VELA	0
492	3107	0	0	0
493	5525	0	LUPUS	0
494	9226	0	0	0
495	9047	0	TELESCOPIUM	0
496	2931	0	0	0
497	4330	D	MONOCEROS	0
498	5532	L	MUSCA	0

4.0	200.80615	55.24805
4.0	133.47884	60.45307
4.0	353.77716	46.18717
4.0	289.69333	17.94296
4.0	109.21525	67.86577
4.0	200.48253	55.18240
4.0	208.80533	41.85738
4.0	307.70584	11.13218
4.0	238.44795	29.06956
4.0	359.18539	6.58650
4.0	68.97215	14.40044
4.0	46.52815	44.66950
4.0	66.42867	15.85282
4.0	325.49374	58.55018
4.0	188.73936	48.26619
4.0	125.79049	3.74207
4.0	61.25546	47.58105
4.0	220.10535	5.44192
4.0	269.53479	2.93239
4.0	225.42570	46.85689
4.0	170.32971	10.80488
4.0	297.08777	70.14071
4.0	43.49563	9.09608
4.0	272.69302	21.07374
4.0	89.40369	42.81705
4.0	49.32280	21.93878
4.0	142.74670	62.56697
4.0	129.53603	35.12983
4.0	47.48646	29.18317
4.0	288.98660	53.27552
4.0	331.17014	25.10020
4.0	187.36331	71.85699
4.0	41.75806	55.68956
4.0	303.01392	46.58886
4.0	55.71148	24.21306
4.0	166.61153	58.70384
4.0	297.48135	0.87590
4.0	336.82721	58.15882
4.0	336.57227	43.75178
4.0	242.71715	63.56031
4.0	233.18114	14.62420
4.1	169.63954	6.30363
4.1	286.51810	37.98430
4.1	131.08286	45.85787
4.1	220.14766	34.95961
4.1	105.37380	15.55800
4.1	341.02988	23.30206
4.1	155.91756	16.58043
4.1	153.15833	41.87367
4.1	338.19669	0.37577
4.1	33.58338	33.61707
4.1	128.97112	42.81330
4.1	158.79874	47.96547
4.1	270.73169	2.50962
4.1	227.11153	48.54920
4.1	119.20044	49.10846
4.1	276.24512	49.10016
4.1	284.75247	5.81101
4.1	107.32758	0.40845
4.1	172.11478	69.60725

499	4560	0	0	0	4.1	29.77980	72.18079
500	9137.	0	0	0	4.1	215.87338	52.08120

YM	CONSTELLATION	STAR NAME	VM	RT ASCEN	DECLIN	RANK	NUM
A	ACHERNAR	-----	0.6	23.96378	- 57.49026	9	2481
A	ANDROMEDA	SIRRAH***	2.1	1.44934	28.81448	51	3765
C	ANDROMEDA	ALMAK	2.3	30.20490	42.09084	64	7734
B	ANDROMEDA	MIRACH	2.4	16.73107	35.35604	73	4471
N	ANDROMEDA	-----	3.1	340.16360	29.95928	158	0734
D	ANDROMEDA	-----	3.5	9.16191	30.58772	249	4058
O	ANDROMEDA	-----	3.6	344.90363	42.05698	287	2609
M	ANDROMEDA	-----	3.9	13.49233	38.22861	384	4281
L	ANDROMEDA	-----	4.0	353.77716	46.18717	441	3204
A	APUS	-----	3.8	220.38747	- 78.83490	375	7193
C	APUS	-----	3.9	246.42825	- 78.78897	394	7407
B	AQUARIUS	-----	3.1	322.23187	- 5.79213	166	5457
A	AQUARIUS	-----	3.2	330.80396	- 0.56350	189	5862
L	AQUARIUS	-----	3.5	342.99960	- 16.08720	241	5375
E	AQUARIUS	-----	3.8	311.24258	- 9.68006	334	4810
L	AQUARIUS	-----	3.8	342.50156	- 7.84615	368	6362
C	AQUARIUS	-----	4.0	334.76840	- 1.63994	432	6044
N	AQUARIUS	-----	4.1	338.19669	- 0.37577	488	6181
A	AQUILA	ALTAIR	0.9	297.08578	8.73492	12	5122
C	AQUILA	TARAZED	2.8	295.97046	10.49011	108	5223
Z	AQUILA	-----	3.0	285.77771	13.78774	138	4461
T	AQUILA	-----	3.4	302.18121	0.97113	229	4150
D	AQUILA	-----	3.4	290.74414	3.01361	233	4603
L	AQUILA	-----	3.5	285.89871	4.95909	247	3021
B	AQUILA	-----	3.9	298.21411	6.28051	400	5235
N	AQUILA	-----	4.0	297.48135	0.87590	475	5159
B	ARA	-----	2.8	260.28442	- 55.48503	103	4725
A	ARA	-----	3.0	261.99313	- 49.83878	149	8069
Z	ARA	-----	3.1	253.61842	- 55.91351	176	4315
N	ARA	-----	3.7	251.36418	- 58.95455	329	4168
D	ARA	-----	3.8	261.64471	- 60.64470	372	3945
A	ARIES	HAMAL	2.2	31.08715	23.22696	58	5151
B	ARIES	CHERATAN	2.7	27.96808	20.56447	99	5012
A	AURIGA	CAPELLA	0.2	78.24777	45.94946	5	0186
B	AURIGA	COLURE	2.1	88.96491	44.94464	50	0750
T	AURIGA	-----	2.7	89.07761	37.21112	96	8636
I	AURIGA	-----	2.9	73.43326	33.08887	125	7522
V	AURIGA	-----	3.3	75.75089	41.16899	210	0026
E	AURIGA	-----	3.5	74.59387	43.75149	248	9955
D	AURIGA	-----	3.9	88.85216	54.28337	383	5502
Z	AURIGA	-----	3.9	74.74450	41.00494	424	9966
A	BOOTES	ARCTURUS	0.2	213.34479	19.44193	6	0944
E	BOOTES	IZAR	2.7	220.70058	27.28407	92	3500
N	BOOTES	-----	2.8	208.07570	18.64760	110	0766
C	BOOTES	-----	3.0	217.51599	38.52615	137	4203
D	BOOTES	-----	3.5	228.37132	33.50033	255	4589
B	BOOTES	-----	3.6	225.01527	40.58689	284	5337
Z	BOOTES	-----	3.9	219.68974	13.94180	412	1145
A	CAELUM	-----	3.8	63.08580	- 42.41676	339	6710
B	CANCER	-----	3.8	123.45117	9.34104	362	6569
A	CANES VENATICI	COR CAROLI	2.9	193.42282	38.58801	127	3257
A	CANIS MAJOR	SIRIUS	-1.6	100.73631	- 16.64621	1	1881
E	CANIS MAJOR	-----	1.6	104.16495	- 28.90284	24	2676
B	CANIS MAJOR	-----	2.0	95.12412	- 17.92978	37	1428
D	CANIS MAJOR	-----	2.0	106.58932	- 26.31260	38	3047
N	CANIS MAJOR	-----	2.4	110.52905	- 29.20446	71	3651
S	CANIS MAJOR	-----	3.1	105.23392	- 23.75902	165	2839
Z	CANIS MAJOR	-----	3.1	94.59811	- 30.03996	175	6698

T	CANIS MAJOR	-----	3.4	100.62069	-	12.95120	216	6074
S	CANIS MAJOR	-----	3.7	104.93159	-	27.86203	299	2797
H	CANIS MAJOR	-----	3.8	101.99301	-	32.44963	367	7258
C	CANIS MAJOR	-----	4.1	105.37380	-	15.55800	484	2303
A	CANIS MINOR	PROCYON	0.5	114.17132		5.35467	8	5756
B	CANIS MINOR	-----	3.1	111.10982		8.39164	177	5456
A	CANOPUS	-----	-0.9	95.71039	-	52.66763	2	4480
D	CAPRICORNUS	-----	3.0	326.07080	-	16.35512	151	4644
B	CAPRICORNUS	-----	3.3	304.55084	-	14.94071	206	3481
A	CAPRICORNUS	-----	3.8	303.82031	-	12.70125	342	3427
C	CAPRICORNUS	-----	3.8	324.33072	-	16.88918	366	4560
Z	CAPRICORNUS	-----	3.9	320.95392	-	22.62898	414	0341
E	CARINA	-----	1.7	125.37243	-	59.34805	27	5932
B	CARINA	-----	1.8	138.16525	-	69.51101	29	0495
1	CARINA	-----	2.3	138.93796	-	59.06491	65	6808
T	CARINA	-----	3.0	160.29189	-	64.13205	148	1083
N	CARINA	-----	3.1	146.46320	-	64.83950	163	0695
Q	CARINA	-----	3.4	153.85242	-	61.08197	217	0905
J	CARINA	-----	3.6	153.13770	-	69.78926	267	0835
G	CARINA	-----	3.6	157.56036	-	61.42769	271	1006
H	CARINA	-----	3.6	118.87678	-	52.84743	288	5635
B	CASSIOPEIA	-----	2.4	1.62390		58.87410	69	1133
A	CASSIOPEIA	SCHEDIR	2.5	9.41380		56.26350	78	1609
D	CASSIOPEIA	-----	2.8	20.63124		59.97621	104	2268
C	CASSIOPEIA	-----	2.8	13.41802		60.44650	105	1482
E	CASSIOPEIA	-----	3.4	27.69374		63.42497	220	2031
N	CASSIOPEIA	-----	3.6	11.51519		57.55087	293	1732
Z	CASSIOPEIA	-----	3.7	8.54305		53.62206	306	1566
A	CENTAURUS	-----	0.1	219.04689	-	60.63023	3	2838
B	CENTAURUS	-----	0.9	210.06879	-	60.13280	11	2582
T	CENTAURUS	-----	2.3	210.91282	-	36.12490	62	5188
C	CENTAURUS	-----	2.4	189.68707	-	48.68537	67	3603
E	CENTAURUS	-----	2.6	204.17635	-	53.21287	82	1047
I	CENTAURUS	-----	2.9	199.44452	-	36.44913	133	4371
D	CENTAURUS	-----	2.9	181.43936	-	50.44398	136	9689
Z	CENTAURUS	-----	3.1	208.10213	-	47.04302	160	4538
M	CENTAURUS	-----	3.3	206.64857	-	42.22546	208	4471
L	CENTAURUS	-----	3.3	173.36595	-	62.74294	214	1472
M	CENTAURUS	-----	3.5	206.62390	-	41.43933	242	4469
M	CENTAURUS	-----	4.0	208.80633	-	41.85738	445	4577
0	CENTAURUS	-----	4.0	166.61153	-	58.70384	474	8813
A	CEPHEUS	ALDERAMIN	2.6	319.34659		62.37326	85	9302
B	CEPHEUS	-----	3.3	322.00552		70.34107	198	0057
C	CEPHEUS	-----	3.4	354.31888		77.35326	237	0818
Z	CEPHEUS	-----	3.6	332.27890		57.95431	289	4137
N	CEPHEUS	-----	3.6	311.06870		61.64413	292	9019
I	CEPHEUS	-----	3.7	341.97336		65.93716	313	0268
M	CEPHEUS	-----	4.0	325.49374		58.55018	452	3693
D	CEPHEUS	-----	4.0	335.82721		58.15882	476	4508
O	CETUS	MIRA	2.0	34.20435	-	3.20372	43	9825
B	CETUS	DENEBKAITOS	2.2	10.27005	-	18.26077	59	7420
A	CETUS	MENKAB	2.8	44.91560	-	3.89476	107	0920
U	CETUS	-----	3.6	25.43608	-	16.20014	262	7986
N	CETUS	-----	3.6	16.51842	-	10.44683	283	7632
C	CETUS	-----	3.6	40.17665	-	3.02612	285	0707
T	CETUS	-----	3.8	20.38068	-	8.44088	338	9274
I	CETUS	-----	3.8	4.21993	-	9.10096	341	8694
Z	CETUS	-----	3.9	27.24770	-	10.58136	408	8059
C	CETUS	-----	4.0	39.22910	-	0.11389	428	0665
A	CIRCINUS	-----	3.4	219.60991	-	64.75900	222	2853

D	CIRCINUS	-----	4.0	242.71715	-	63.56031	478	3474
A	COLUMBA	-----	2.8	84.45921	-	34.09965	101	6059
B	COLUMBA	-----	3.2	87.29887	-	35.78604	187	6240
E	COLUMBA	-----	3.9	82.35302	-	35.50611	427	5924
H	COLUMBA	-----	4.0	95.07116	-	33.41002	436	6735
N	COLUMBA	-----	4.0	89.40369	-	42.81705	463	7650
B	CORONA AUSTRALIS	-----	4.1	286.51810	-	37.98430	481	0990
E	CORONA BOREALIS	-----	2.0	239.35210		26.06084	40	4129
A	CORONA BOREALIS	GEMMA	2.3	233.14227		26.88188	63	3893
B	CORONA BOREALIS	-----	3.7	231.44144		29.27702	319	3831
D	CORONA BOREALIS	-----	3.9	235.16026		26.45288	415	3958
B	CORVUS	-----	2.8	187.93892	-	23.12045	102	0915
C	CORVUS	-----	2.8	183.30768	-	17.26440	114	7176
D	CORVUS	ALGORAB	3.1	186.81825	-	16.23726	174	7323
E	CORVUS	-----	3.2	181.88716	-	22.34170	186	0531
D	CRATER	-----	3.4	167.90443		15.70316	221	9512
C	CRUX	-----	1.6	187.09470	-	56.83342	23	0019
A	CYGNUS	DENEB	1.3	309.93143		45.10086	17	9941
C	CYGNUS	-----	2.3	305.10803		40.09571	61	9528
E	CYGNUS	-----	2.6	311.04654		33.78199	84	0474
D	CYGNUS	-----	3.0	295.85278		45.00784	140	8796
B	CYGNUS	ALBIREO	3.2	292.17593		27.85346	191	7301
Z	CYGNUS	-----	3.4	317.70148		30.02098	227	1070
F	CYGNUS	-----	3.9	315.77747		43.72759	385	0424
I	CYGNUS	-----	3.9	292.11115		51.62249	401	1702
N	CYGNUS	-----	4.0	298.60718		34.94951	438	9116
B	DELPHINUS	-----	3.7	308.80087		14.42009	301	6316
A	DELPHINUS	-----	3.9	309.32864		15.73453	396	6357
E	DELPHINUS	-----	4.0	307.70584		11.13218	446	6230
A	DORADO	-----	3.5	68.22832	-	55.14772	260	3564
B	DORADO	-----	3.8	83.29721	-	62.52228	347	9311
C	DRACO	ETAMIN	2.4	268.86075		51.49404	68	0653
N	DRACO	-----	2.9	245.82698		61.62697	129	7074
E	DRACO	-----	3.0	262.32504		52.33768	144	0429
D	DRACO	-----	3.2	288.13678		67.57362	180	8222
Z	DRACO	-----	3.2	257.15894		65.77612	194	7365
I	DRACO	-----	3.5	230.95309		59.14060	256	9520
A	DRACO	-----	3.6	210.75818		64.61433	269	6273
H	DRACO	-----	3.7	275.48950		72.71171	323	9087
H	DRACO	-----	3.9	187.83980		70.06361	392	7593
F	DRACO	-----	3.9	268.16541		56.87991	407	0631
E	DRACO	-----	4.0	297.08777		70.14071	460	9540
B	ERIDANUS	-----	2.9	76.34735	-	5.14959	130	1794
C	ERIDANUS	-----	3.2	58.92363	-	13.64949	192	9283
T	ERIDANUS	-----	3.4	44.09118	-	40.50421	240	6113
U4	ERIDANUS	-----	3.6	64.00015	-	33.91935	273	4902
D	ERIDANUS	-----	3.7	55.21265	-	9.93140	309	0686
H	ERIDANUS	-----	3.7	28.50343	-	51.85715	311	2573
E	ERIDANUS	-----	3.8	51.11387	-	9.55977	349	1195
M	ERIDANUS	-----	3.8	72.13509	-	5.52121	360	2142
R	ERIDANUS	-----	3.8	33.68092	-	51.74303	364	2696
E	ERIDANUS	-----	3.8	52.64315	-	9.62632	370	0564
O	ERIDANUS	-----	3.9	60.12327	-	5.85208	398	1579
0	ERIDANUS	-----	3.9	55.44700	-	42.42240	404	9078
V	ERIDANUS	-----	3.9	68.40131	-	30.66360	423	5148
A	FORNAX	-----	4.0	47.48646	-	29.18317	467	8373
B	GEMINI	POLLUX	1.2	115.56465		28.14864	16	9666
A	GEMINI	CASTOR	1.6	112.85272		31.99974	22	0198
C	GEMINI	ALHENA	1.9	98.70582		16.44372	34	5912
E	GEMINI	MEBSUTA	3.2	100.21401		25.18248	188	8682

M	GEMINI	-----	3.2	94.98373	22.54111	193	8297
D	GEMINI	-----	3.5	109.28444	22.07613	244	9294
T	GEMINI	-----	3.6	102.37363	34.02349	265	9570
C	GEMINI	-----	3.6	108.80499	16.63226	295	6746
N	GEMINI	-----	3.7	92.96461	22.52316	305	8135
H	GEMINI	-----	3.7	115.35773	24.51958	314	9653
Z	GEMINI	-----	3.9	105.28594	20.64540	413	9031
I	GEMINI	-----	3.9	110.65573	27.89921	417	9374
A	GRUS	-----	2.2	331.27286	- 47.20404	57	0992
B	GRUS	-----	2.2	339.92267	- 47.14672	60	1258
C	GRUS	-----	3.2	327.72687	- 37.60099	182	3374
E	GRUS	-----	3.7	341.38629	- 51.58035	300	7593
D	GRUS	-----	4.0	336.57227	- 43.75178	477	1154
B	HERCULES	-----	2.8	247.01711	21.59727	112	4411
Z	HERCULES	-----	3.0	249.84995	31.69226	142	5485
D	HERCULES	-----	3.2	258.24399	24.89688	181	4951
P	HERCULES	-----	3.4	258.32590	36.86440	218	5890
M	HERCULES	-----	3.5	266.12500	27.74872	245	5397
N	HERCULES	-----	3.6	250.29491	39.01633	264	5504
I	HERCULES	-----	3.8	264.51276	46.03202	337	6872
F	HERCULES	-----	3.8	268.95505	29.25204	346	5590
E	HERCULES	-----	3.9	254.59363	30.99887	402	5716
U	HERCULES	-----	3.9	244.55878	46.43156	421	6028
T	HERCULES	-----	4.0	268.63409	37.25602	430	6485
A	HYDRA	ALPHARD	2.2	141.28246	- 8.44096	53	6871
C	HYDRA	-----	3.3	199.04945	- 22.90830	202	1543
Z	HYDRA	-----	3.3	133.18788	- 6.13701	211	7264
V	HYDRA	-----	3.3	161.78888	- 15.93151	212	6256
C	HYDRA	-----	3.5	260.29471	- 56.33301	254	4726
E	HYDRA	-----	3.5	131.03232	- 6.60345	257	7112
P	HYDRA	-----	3.5	210.87935	- 26.44236	258	2244
F	HYDRA	-----	3.7	172.63443	- 31.58074	307	2558
T	HYDRA	-----	3.8	137.94090	- 2.52629	314	7527
I	HYDRA	-----	3.8	152.03728	- 12.10626	369	5785
B	HYDRUS	-----	2.9	5.78906	- 77.53569	126	5670
A	HYDRUS	-----	3.0	29.29876	- 61.81262	147	8474
C	HYDRUS	-----	3.2	56.99777	- 74.39265	183	6029
A	INDUS	-----	3.2	308.51471	- 47.46748	185	0300
B	INDUS	-----	3.7	312.72949	- 58.64450	315	6784
A	LACERTA	-----	3.9	337.30649	50.02499	405	4542
A	LEO	REGULUS	1.3	151.42769	12.21237	19	8967
B	LEO	DENEBOLA	2.2	176.62752	14.85161	55	9809
C	LEO	ALGEIBA	2.6	154.30467	20.09524	83	1298
D	LEO	ZOSMA	2.6	167.86285	20.79796	86	1727
E	LEO	-----	3.1	145.75415	24.00545	171	1004
Z	LEO	-----	3.6	153.47816	23.66721	266	1265
N	LEO	-----	3.6	151.15222	17.00727	275	8955
C	LEO	-----	3.8	154.30585	20.09436	303	1299
O	LEO	-----	3.8	144.62086	10.12073	379	8709
Y	LEO	-----	3.9	157.54485	9.56451	387	8355
I	LEO	-----	4.0	170.32971	10.80488	459	9587
S	LEO	-----	4.0	233.18114	14.62420	479	9370
O	LEO MINOR	-----	3.9	162.63013	34.48489	419	2297
A	LEPUS	ARNEB	2.7	82.63081	- 17.85672	95	0547
B	LEPUS	NIHAL	3.0	81.52538	- 20.79803	146	0457
M	LEPUS	-----	3.3	77.67089	- 16.26332	200	0237
E	LEPUS	-----	3.3	75.83568	- 22.43690	215	0051
Z	LEPUS	-----	3.7	86.17207	- 14.83922	303	0801
D	LEPUS	-----	3.8	85.59439	- 22.46342	344	0759
N	LEPUS	-----	3.8	88.53147	- 14.17546	365	0957

D	LEPUS	-----	3.9	87.29247	-	20.88201	395	0926
B	LIBRA	-----	2.7	228.57805	-	9.19970	93	0430
A	LIBRA	-----	2.9	222.02686	-	15.63516	119	8840
S	LIBRA	-----	3.4	225.28436	-	25.08674	230	3139
S	LIBRA	-----	3.5	201.74367	-	23.02348	253	1695
C	LIBRA	-----	4.1	169.63954	-	6.30363	480	8804
N	LUPUS	-----	2.6	218.08044	-	41.93938	87	5044
B	LUPUS	-----	2.8	223.81161	-	42.93391	109	5335
A	LUPUS	-----	2.9	219.64804	-	47.17481	131	5128
Z	LUPUS	-----	3.0	232.94995	-	41.00027	143	5938
D	LUPUS	-----	3.4	223.97462	-	41.90491	219	5344
E	LUPUS	-----	3.4	229.52022	-	40.46803	224	5691
A	LUPUS	-----	3.5	227.16983	-	51.91050	252	2304
R	LUPUS	-----	3.6	229.65630	-	36.08144	270	6552
E	LUPUS	-----	3.7	229.81880	-	44.51117	298	5712
V	LUPUS	-----	3.8	233.49568	-	27.97094	356	3619
P	LUPUS	-----	4.0	225.42570	-	46.85689	458	5426
0	LUPUS	-----	4.1	227.11153	-	48.54920	493	5525
A	LYNX	-----	3.3	139.50360		34.60516	199	1414
0	LYNX	-----	3.8	138.93177		37.01543	373	1391
A	LYRA	VEGA	0.1	278.81107		38.73602	4	7174
C	LYRA	-----	3.3	284.26788		32.61979	205	7663
B	LYRA	-----	3.9	282.05807		33.30347	403	7451
F	MICROSCOPIUM	-----	3.8	318.19815		37.83119	357	1121
0	MONOCEROS	-----	4.0	125.79049	-	3.74207	454	5896
D	MONOCEROS	-----	4.1	107.32758	-	0.40845	497	4330
A	MUSCA	-----	2.9	188.54456	-	68.86023	117	1974
B	MUSCA	-----	3.3	190.79788	-	67.83477	209	2019
D	MUSCA	-----	3.6	194.69989	-	71.27978	282	7000
L	MUSCA	-----	3.8	175.80843	-	66.45140	340	1575
C	MUSCA	-----	4.0	187.36331	-	71.85699	470	6955
L	MUSCA	-----	4.1	172.11478	-	69.60725	498	5532
N	NORMA	-----	3.6	239.20010	-	38.25555	296	7208
V	OCTANS	-----	3.7	323.99945	-	77.61395	331	7948
A	OPHIUCHUS	RAS ALHAGUE	2.1	263.15289		12.59498	49	2932
N	OPHIUCHUS	-----	2.6	256.87692	-	15.66469	89	0332
Z	OPHIUCHUS	-----	2.7	248.60057	-	10.46745	94	0006
B	OPHIUCHUS	-----	2.9	265.25018		4.58662	116	2671
D	OPHIUCHUS	-----	3.0	242.93053	-	3.56706	156	1052
E	OPHIUCHUS	-----	3.3	243.91812	-	4.57210	213	1086
T	OPHIUCHUS	-----	3.4	259.73398	-	24.95144	225	5320
H	OPHIUCHUS	-----	3.4	253.82483		9.45134	236	1962
N	OPHIUCHUS	-----	3.5	258.09146		14.44593	246	2680
M	OPHIUCHUS	-----	3.6	236.75186	-	3.27860	274	0787
0	OPHIUCHUS	-----	3.7	271.24438		9.55526	304	3142
C	OPHIUCHUS	-----	3.7	266.34592		2.72452	326	2754
0	OPHIUCHUS	-----	4.0	269.53479		2.93239	457	3013
B	ORION	RIGEL	0.3	78.03329	-	8.25796	7	1907
A	ORION	BETELGEUSE	0.6	88.11587		7.39942	10	3271
C	ORION	BELLATRIX	1.7	80.61179		6.30603	28	2740
D	ORION	MINTAKA	1.8	83.41865	-	1.23231	30	2346
E	ORION	ALNILAM	2.0	84.55951	-	1.96749	39	2444
K	ORION	SAIPH	2.2	86.34589	-	9.68596	54	2542
D	ORION	-----	2.5	82.36257	-	0.33456	80	2220
I	ORION	-----	2.9	83.24636	-	5.94118	132	2323
P4	ORION	-----	3.3	71.78079		6.87563	207	2106
N	ORION	-----	3.4	80.49032	-	2.44155	228	2071
U	ORION	-----	3.7	78.79417	-	6.89696	317	1952
R	ORION	-----	3.7	83.09547	-	9.90232	332	2921
S	ORION	-----	3.8	84.05853	-	2.62736	376	2406

P5	ORION	-----	3.9	72.91109		2.36034	420	2197
A	PAVO	-----	2.1	305.42621	-	56.89723	46	6574
D	PAVO	-----	3.6	300.96021	-	66.31207	272	4733
N	PAVO	-----	3.6	265.20517	-	64.70274	278	4020
B	PAVO	-----	3.6	310.11942	-	66.38483	290	4862
E	PEGASUS	ENIT	2.5	325.43231		9.64491	76	7029
A	PEGASUS	MARKAB	2.6	345.56693		14.93586	88	8378
B	PEGASUS	SCHEAT	2.6	345.33655		27.81124	91	0981
C	PEGASUS	-----	2.9	2.66436		14.90571	134	1781
Z	PEGASUS	-----	3.6	339.74164		10.56983	263	8103
T	PEGASUS	-----	3.7	331.91916		5.95115	302	7340
B	PEGASUS	SCHEAT	3.7	341.89664		24.33714	310	0816
I	PEGASUS	-----	4.0	331.17014		25.10020	469	0238
L	PEGASUS	-----	4.1	341.02988		23.30206	485	0775
A	PERSEUS	MIRFAK	1.9	50.18517		49.68501	32	8787
B	PERSEUS	ALGOL	2.9	46.22648		40.76457	123	8592
Z	PERSEUS	-----	2.9	57.74566		31.73682	128	6799
E	PERSEUS	-----	3.0	58.62263		39.86737	155	6840
C	PERSEUS	-----	3.1	45.28998		53.31230	170	3789
D	PERSEUS	-----	3.1	54.83850		47.62954	172	9053
Y	PERSEUS	-----	3.7	45.49081		38.64802	312	6138
F	PERSEUS	-----	4.0	58.92844		35.64903	429	6856
N	PERSEUS	-----	4.0	41.75806		55.68956	471	3655
A	PHOENIX	-----	2.4	5.95429	-	42.57737	74	5093
B	PHOENIX	-----	3.4	15.96399	-	46.98605	226	5365
C	PHOENIX	-----	3.4	21.54907	-	43.57387	233	5516
E	PHOENIX	-----	3.9	1.71990	-	46.02325	389	4983
D	PHOENIX	-----	4.0	22.29303	-	49.33200	434	5536
A	PICTOR	-----	3.3	101.91969	-	61.88733	204	9647
B	PICTOR	-----	3.9	86.52473	-	51.08382	425	4134
N	PISCES	-----	3.7	22.20082		15.08872	297	2484
C	PISCES	-----	3.9	348.64304		3.00879	393	8085
J	PISCES	-----	4.0	359.18539		6.58650	448	8513
A	PISCIS AUSTRINUS	FORMALHAUT	1.3	343.72305	-	29.88772	18	1524
N	PLEIADS	ALCYONE	3.0	56.12677		23.95210	154	6199
0	PLEIADS	-----	4.0	55.71148		24.21306	473	6155
B	POLARIS VOLAN	-----	3.6	126.29981	-	65.96972	281	0228
C	POLARIS VOLAN	-----	3.9	107.29501	-	70.41798	410	6374
Z	POLARIS VOLAN	-----	3.9	115.61243	-	72.48630	416	6438
D	POLARIS VOLAN	-----	4.0	109.21525	-	67.86577	443	9809
Z	PUPPIS	-----	2.3	120.45647	-	39.86134	66	8752
P	PUPPIS	-----	2.7	103.84396	-	37.00664	97	7795
U	PUPPIS	-----	2.8	102.17368	-	50.55430	100	4735
P	PUPPIS	-----	2.9	121.35342	-	24.15897	124	5217
S	PUPPIS	-----	3.1	108.00286	-	44.55733	167	8549
V	PUPPIS	-----	3.2	99.05748	-	43.15114	190	8071
S	PUPPIS	-----	3.3	111.91090	-	43.19929	203	8755
F	PUPPIS	-----	3.5	116.79758	-	24.73312	259	4601
A	PUPPIS	-----	3.8	117.62428	-	40.44591	345	9082
A	PYXIS	-----	3.7	130.39537	-	33.00518	318	9546
B	PYXIS	-----	4.0	129.53603	-	35.12983	466	9490
A	RECTICULUM	-----	3.4	63.44393	-	62.59862	234	8969
B	RECTICULUM	-----	3.8	55.89151	-	64.96394	333	8877
C	SAGITTA	-----	3.7	299.13309		19.35513	330	5500
D	SAGITTA	-----	3.8	296.28928		18.40964	358	5259
E	SAGITTARIUS	-----	2.0	275.21329	-	34.41026	42	0091
S	SAGITTARIUS	-----	2.1	283.04135	-	26.36064	48	7448
-	SAGITTARIUS	-----	2.7	284.85788	-	29.95355	98	7600
D	SAGITTARIUS	-----	2.8	274.44815	-	29.85130	111	6681
L	SAGITTARIUS	-----	2.9	276.22107	-	25.45116	121	6841

P	SAGITTARIUS	-----	3.0	286.69772	-	21.10486	139	7756
C	SAGITTARIUS	-----	3.1	270.64865	-	30.42661	159	9696
N	SAGITTARIUS	-----	3.2	273.56073	-	36.77875	195	9957
R	SAGITTARIUS	-----	3.3	280.63318	-	27.04402	201	7239
U	SAGITTARIUS	-----	3.4	285.95483	-	27.74532	232	7683
F	SAGITTARIUS	-----	3.6	283.68680	-	21.17413	268	7504
M	SAGITTARIUS	-----	4.0	272.69302	-	21.07374	462	6497
A	SCORPIUS	ANTARES	1.1	246.58420	-	26.32277	13	4415
C	SCORPIUS	-----	1.7	262.55261	-	37.06934	26	8954
T	SCORPIUS	-----	2.0	263.43073	-	42.96810	41	8201
E	SCORPIUS	-----	2.4	251.73003	-	34.20432	72	8078
K	SCORPIUS	-----	2.5	264.75644	-	39.00632	77	9163
D	SCORPIUS	-----	2.5	239.34299	-	22.48094	81	4014
L	SCORPIUS	-----	2.8	261.84027	-	37.25793	115	8896
B	SCORPIUS	AKRAB	2.9	240.63129	-	19.67012	120	9682
T	SCORPIUS	-----	2.9	248.19127	-	28.11404	122	4481
P	SCORPIUS	-----	3.0	238.95541	-	25.97171	153	3987
S	SCORPIUS	-----	3.1	244.53612	-	25.47448	157	4336
I	SCORPIUS	-----	3.1	266.02124	-	40.10971	173	8420
M	SCORPIUS	-----	3.1	252.11949	-	37.96359	179	8102
G	SCORPIUS	-----	3.3	266.61322	-	37.02934	197	9318
N	SCORPIUS	-----	3.4	257.14172	-	43.17531	235	7707
M	SCORPIUS	-----	3.6	252.23582	-	37.93422	277	8116
Z	SCORPIUS	-----	3.8	252.76477	-	42.27774	350	7402
0	SCORPIUS	-----	4.0	238.44795	-	29.06956	447	3957
A	SERPENS CAPUT	UNULA	2.8	235.45062		6.58165	113	1157
B	SERPENS CAPUT	-----	3.7	235.96945		15.57706	325	1725
E	SERPENS CAPUT	-----	3.8	237.08018		4.62690	359	1218
L	SERPENS CAPUT	-----	3.9	247.09726		2.09196	386	1658
C	SERPENS CAPUT	-----	3.9	238.53523		15.82355	399	1826
V	SERPENS CAUDA	-----	3.4	274.68027		2.91339	223	2241
V	SERPENS CAUDA	-----	3.5	269.06805		9.76919	250	2004
F	SERPENS CAUDA	-----	3.6	263.68030		15.36884	286	0700
B	SOUTHERN CROSS	-----	1.5	191.19598	-	59.41570	20	0259
A	SOUTHERN CROSS	-----	1.6	185.95027	-	62.82206	21	1904
E	SOUTHERN CROSS	-----	3.1	183.11917	-	58.47078	178	9791
E	SOUTHERN CROSS	-----	3.6	184.66174	-	60.12502	279	1862
A	TAURUS	ALDERBARON	1.1	68.26207		16.41042	14	4027
B	TAURUS	NATH	1.8	80.78213		28.56715	31	7168
Z	TAURUS	-----	3.0	83.66360		21.11389	150	7336
E	TAURUS	-----	3.6	66.42327		19.07121	276	3954
T	TAURUS	-----	3.6	66.45092		15.76161	280	3957
O	TAURUS	-----	3.8	50.52967		8.85424	374	1172
C	TAURUS	-----	3.9	64.23605		15.50851	411	3868
D	TAURUS	-----	3.9	65.01185		17.42689	426	3897
L	TAURUS	-----	4.0	59.47657		12.35059	433	3719
C	TAURUS	-----	4.0	66.42867		15.85282	451	3955
A	TELESCOPIUM	-----	3.8	275.81656	-	45.99809	355	9023
T	TELESCOPIUM	-----	3.9	270.68430	-	50.09709	418	5242
0	TELESCOPIUM	-----	4.1	276.24512	-	49.10016	495	9047
B	TRIANGULUM	-----	3.1	31.63991		34.75181	169	5306
A	TRIANGULUM	-----	3.6	27.55600		29.33615	261	4996
C	TRIANGULUM	-----	4.1	33.58338		33.61707	489	5427
A	TRIANGULUMAUSTRALE	-----	1.9	250.83775	-	68.93886	35	3700
B	TRIANGULUMAUSTRALE	-----	3.0	237.67908	-	63.27850	145	3346
C	TRIANGULUMAUSTRALE	-----	3.1	228.55244	-	68.49689	162	3097
A	TUCANA	-----	2.9	333.77371	-	60.50973	118	5193
E	URSA MAJOR	ALIOTH	1.7	192.95868		56.23087	25	8553
N	URSA MAJOR	BENETNASCH	1.9	206.39290		49.56226	33	4752
A	URSA MAJOR	DUBHE	2.0	165.16487		62.02133	45	5384

Z	URSA MAJOR	MIZAR	2.4	200.47882	55.18596	70	8737
B	URSA MAJOR	MERAK	2.4	164.70938	56.65087	75	7876
C	URSA MAJOR	PHEODA	2.5	177.80235	53.97277	79	8179
W	URSA MAJOR	-----	3.1	166.71494	44.77017	161	3629
I	URSA MAJOR	-----	3.1	133.94844	48.23946	168	2630
M	URSA MAJOR	-----	3.2	154.83945	41.75174	184	3310
T	URSA MAJOR	OWL NEBULA	3.3	142.38121	51.90644	196	7289
D	URSA MAJOR	-----	3.4	183.23993	57.31026	239	8315
L	URSA MAJOR	-----	3.5	153.52231	43.16488	243	3268
O	URSA MAJOR	-----	3.5	126.53180	60.88738	251	4573
V	URSA MAJOR	-----	3.7	168.94562	33.36737	316	2486
H	URSA MAJOR	-----	3.7	135.05528	47.35577	321	2661
V	URSA MAJOR	-----	3.9	168.87984	31.81080	391	2484
H	URSA MAJOR	-----	3.9	175.85429	48.05671	406	3886
V	URSA MAJOR	-----	3.9	146.86293	59.27509	409	7401

```
C   ************************************************************************
C   *    PROGRAM DUMBO                                                     *
C   *              DETERMINING URANOGRAPHIC MATCHES from                   *
C   *                        BANDS OBSERVED                                *
C   *                                                                      *
C   *    WRITTEN BY:    LAURA STERRETT                                     *
C   *    MODIFIED BY:   TIM POTH & NICKI COOK                              *
C   *                                                                      *
C   *    PURPOSE:   THIS PROGRAM READS IN THE SAO, PRIMARY, AND SUPPLEMENT- *
C   *               ARY MAPS AND MATCHES STARS, BASED ON LOCATION, VISUAL   *
C   *               MAGNITUDE, AND SPECTRAL TYPE.  ADDITIONALLY, THIS       *
C   *               COMPUTES THE PEAK IRRADIANCE MAGNITUDES FOR FURTHER     *
C   *               COMPARISON AND EVALUATION.                              *
C   *                                                                      *
C   *    DATE:      JULY 5,1990                                            *
C   ************************************************************************


C  This allows the SAO map to be read in by array of up to 30000 elements

C         COMMON/ERR/XX
          DIMENSION SRA(30000),SDEC(30000),B(30000)
          DIMENSION SND(30000),PM(30000)
          DIMENSION VMAG(30000),DV(30000)
          DIMENSION PMA(30000),CI1(30000),CI2(30000),SRAR(30000)
          DIMENSION SDECR(30000)
          CHARACTER*1 SPECT(30000),SUBSP(30000),VMA(30000)
          INTEGER SAON(30000),FILECNT,XX
          CHARACTER*20 SAO,PRI,SUP,OUT,BAND,DEND
          REAL*4 VMAG,IRMAG,TEMP,IRM4,IRM11,IRM20,IRM27,PM,TRAD
          CHARACTER*2 EM4,EM11,EM20,EM27,ANS

C    The following commom statements are used in subroutine "ANGLE"

          COMMON/GL/ RA,DEC,IRM4,EM4,IRM11,EM11,IRM20,EM20,IRM27,EM27
          COMMON/NUM/ NSAO,NPRI,NSUP,NOUT,NBAND
          COMMON/TREE/JCOUNT

C    This defines the files which are opened

         DO 777 FILECNT=1,18
          IF(FILECNT.EQ.1)THEN
            SAO= '[STARS]SAO85.DAT'
            PRI= '[STARS]PRI85.DAT'
            SUP= '[STARS]SUP85.DAT'
            OUT= 'POSMATCH85.DAT'
            DEND='MAGMATCH85.DAT'
            GO TO 1
          ENDIF
           IF(FILECNT.EQ.2)THEN
            SAO= '[STARS]SAO75.DAT'
            PRI= '[STARS]PRI75.DAT'
            SUP= '[STARS]SUP75.DAT'
            OUT= 'POSMATCH75.DAT'
            DEND='MAGMATCH75.DAT'
            GO TO 1
          ENDIF
           IF(FILECNT.EQ.3)THEN
            SAO= '[STARS]SAO65.DAT'
            PRI= '[STARS]PRI65.DAT'
```

```
      SUP= '[STARS]SUP65.DAT'
      OUT= 'POSMATCH65.DAT'
      DEND='MAGMATCH65.DAT'
      GO TO 1
     ENDIF
      IF(FILECNT.EQ.4)THEN
       SAO= '[STARS]SAO55.DAT'
       PRI= '[STARS]PRI55.DAT'
       SUP= '[STARS]SUP55.DAT'
       OUT= 'POSMATCH55.DAT'
       DEND='MAGMATCH55.DAT'
       GO TO 1
     ENDIF
      IF(FILECNT.EQ.5)THEN
       SAO= '[STARS]SAO45.DAT'
       PRI= '[STARS]PRI45.DAT'
       SUP= '[STARS]SUP45.DAT'
       OUT= 'POSMATCH45.DAT'
       DEND='MAGMATCH45.DAT'
       GO TO 1
     ENDIF
      IF(FILECNT.EQ.6)THEN
       SAO= '[STARS]SAO35.DAT'
       PRI= '[STARS]PRI35.DAT'
       SUP= '[STARS]SUP35.DAT'
       OUT= 'POSMATCH35.DAT'
       DEND='MAGMATCH35.DAT'
       GO TO 1
     ENDIF
      IF(FILECNT.EQ.7)THEN
       SAO= '[STARS]SAO25.DAT'
       PRI= '[STARS]PRI25.DAT'
       SUP= '[STARS]SUP25.DAT'
       OUT= 'POSMATCH25.DAT'
       DEND='MAGMATCH25.DAT'
       GO TO 1
     ENDIF
      IF(FILECNT.EQ.8)THEN
       SAO= '[STARS]SAO15.DAT'
       PRI= '[STARS]PRI15.DAT'
       SUP= '[STARS]SUP15.DAT'
       OUT= 'POSMATCH15.DAT'
       DEND='MAGMATCH15.DAT'
       GO TO 1
     ENDIF
      IF(FILECNT.EQ.9)THEN
       SAO= '[STARS]SAO05.DAT'
       PRI= '[STARS]PRI05.DAT'
       SUP= '[STARS]SUP05.DAT'
       OUT= 'POSMATCH05.DAT'
       DEND='MAGMATCH05.DAT'
       GO TO 1
      ENDIF
      IF(FILECNT.EQ.10)THEN
       SAO= '[STARS]SAO-05.DAT'
       PRI= '[STARS]PRI-05.DAT'
       SUP= '[STARS]SUP-05.DAT'
       OUT= 'POSMATCH-05.DAT'
       DEND='MAGMATCH-05.DAT'
       GO TO 1
```

```
 ENDIF
  IF(FILECNT.EQ.11)THEN
   SAO= '[STARS]SAO-15.DAT'
   PRI= '[STARS]PRI-15.DAT'
   SUP= '[STARS]SUP-15.DAT'
   OUT= 'POSMATCH-15.DAT'
   DEND='MAGMATCH-15.DAT'
   GO TO 1
 ENDIF
  IF(FILECNT.EQ.12)THEN
   SAO= '[STARS]SAO-25.DAT'
   PRI= '[STARS]PRI-25.DAT'
   SUP= '[STARS]SUP-25.DAT'
   OUT= 'POSMATCH-25.DAT'
   DEND='MAGMATCH-25.DAT'
   GO TO 1
 ENDIF
  IF(FILECNT.EQ.13)THEN
   SAO= '[STARS]SAO-35.DAT'
   PRI= '[STARS]PRI-35.DAT'
   SUP= '[STARS]SUP-35.DAT'
   OUT= 'POSMATCH-35.DAT'
   DEND='MAGMATCH-35.DAT'
   GO TO 1
 ENDIF
  IF(FILECNT.EQ.14)THEN
   SAO= '[STARS]SAO-45.DAT'
   PRI= '[STARS]PRI-45.DAT'
   SUP= '[STARS]SUP-45.DAT'
   OUT= 'POSMATCH-45.DAT'
   DEND='MAGMATCH-45.DAT'
   GO TO 1
 ENDIF
  IF(FILECNT.EQ.15)THEN
   SAO= '[STARS]SAO-55.DAT'
   PRI= '[STARS]PRI-55.DAT'
   SUP= '[STARS]SUP-55.DAT'
   OUT= 'POSMATCH-55.DAT'
   DEND='MAGMATCH-55.DAT'
   GO TO 1
 ENDIF
  IF(FILECNT.EQ.16)THEN
   SAO= '[STARS]SAO-65.DAT'
   PRI= '[STARS]PRI-65.DAT'
   SUP= '[STARS]SUP-65.DAT'
   OUT= 'POSMATCH-65.DAT'
   DEND='MAGMATCH-65.DAT'
   GO TO 1
 ENDIF
  IF(FILECNT.EQ.17)THEN
   SAO= '[STARS]SAO-75.DAT'
   PRI= '[STARS]PRI-75.DAT'
   SUP= '[STARS]SUP-75.DAT'
   OUT= 'POSMATCH-75.DAT'
   DEND='MAGMATCH-75.DAT'
   GO TO 1
 ENDIF
  IF(FILECNT.EQ.18)THEN
   SAO= '[STARS]SAO-85.DAT'
   PRI= '[STARS]PRI-85.DAT'
```

```
          SUP= '[STARS]SUP-85.DAT'
          OUT= 'POSMATCH-85.DAT'
          DEND='MAGMATCH-85.DAT'
         ELSE
          WRITE(*,*)'FILECNT STOP'
          STOP
        ENDIF

C   Defining File Numbers

        DATA NSAO,NPRI,NSUP,NOUT,NEND /10,20,30,40,50/
 1      OPEN(NSAO,FILE=SAO,STATUS='OLD',READONLY)
        OPEN(NPRI,FILE=PRI,STATUS='OLD',READONLY)
        OPEN(NSUP,FILE=SUP,STATUS='OLD',READONLY)
        OPEN(NOUT,FILE=OUT,STATUS='NEW')
        OPEN(NEND,FILE=DEND,STATUS='NEW')
        OPEN(6,FILE='SS_STAR_CAT.DAT',STATUS='NEW')

C   Initializing counter for star matches

       JCOUNT=0

C   Defining constants

        DATA PI, C1, C2 /3.141592654, 3.7415E+04, 1.43879E+04/

C   This loop reads in the SAO map in an array for comparison

        WRITE(*,*)' READING IN SAO MAP...',FILECNT
        DO 299 II=1,27000
         READ(NSAO,25,END=299) SAON(II),B(II),SRA(II),SND(II),
     +    SDEC(II),PM(II),VMAG(II),SPECT(II),SUBSP(II),DV(II),
     +    VMA(II),PMA(II),CI1(II),CI2(II),SRAR(II),SDECR(II)
 25      FORMAT(I7,A1,F10.5,A1,F11.5,1X,F4.1,1X,F4.1,A1,A1,1X,A1,
     +    A1,A1,A2,A1,F10.8,F11.8)
        ICNT=II
 299    CONTINUE

C   To check SAO completion

        WRITE(*,*)' READING SAO MAP COMPLETE...'

C   This initializes the counter which checks for duplicated matches

         ICOUNT=0
         NCOUNT=0

C   Read in AFGL Map ( Primary and Secondary )

       WRITE(*,*)' CHECKING AFGL PRIMARY MAP FOR POSITION MATCH...'
        DO 11 JDEC=1,1000
         READ(NPRI,35,END=11)NAFGL,AFGLN,RA,DEC,IRM4,EM4,
     &    IRM11,EM11,IRM20,EM20,IRM27,EM27

C   This calls  "ANGLE" subroutine in order to check right ascension and
C   declination matches

         CALL ANGLE(SRA,SDEC,VMAG,VMA,SPECT,SUBSP,ICNT)
 11      CONTINUE
```

```
C   This loop compares the supplementary map with the SAO map

      WRITE(*,*)' CHECKING SUPPLEMENTAL MAP FOR POSITION MATCH...'
         DO 33 JO =1,1000
          READ(NSUP,35,END=33)NAFGL,AFGLN,RA,DEC,IRM4,EM4,
     &     IRM11,EM11,IRM20,EM20,IRM27,EM27
          CALL ANGLE(SRA,SDEC,VMAG,VMA,SPECT,SUBSP,ICNT)
 35       FORMAT(I5,2X,A5,F10.5,3X,F10.5,1X,F4.1,2X,A2,2X,
     &     F4.1,2X,A2,2X,F4.1,2X,A2,2X,F4.1,2X,A2)
 33      CONTINUE

C   This notes the finish of the comparisen of the supplmental and SAO
C   maps

        WRITE(*,*)' POSITION MATCHING COMPLETE FOR BOTH IR MAPS...'

C   Calculating the actual temperature of the star from the spectral
C   class and subclass

         WRITE(*,*)' CHECKING FOR MATCHES IN MAGNITUDE...'

         REWIND (NOUT)
         DO 44 J=1,1000
          READ(NOUT,45,END=99)ICOUNT,JCOUNT,SRA(J),RA,SDEC(J),
     &     DEC,VMAG(J),VMA(J),SPECT(J),SUBSP(J),IRM4,EM4,IRM11,
     &     EM11,IRM20,EM20,IRM27,EM27
 45       FORMAT(I6,1X,I6,1X,F10.5,1X,F10.5,F11.5,F10.5,F4.1,1X,
     &     A1,A1,A1,1X,F4.1,1X,A2,1X,F4.1,1X,A2,1X,F4.1,1X,A2,1X,
     &     F4.1,1X,A2)

          CALL TEMPER(SPECT,SUBSP,T,J)
          IF (T .EQ. 0.) GO TO 44

C   Checking for zero readings in the IR wavebands and assigning
C      wavelengths

 191       IF ( IRM4 .LT. 99.9)THEN
           NLAM = 1
           IRMAG = IRM4
          IJ=1
          ELSE
          IF (IRM11 .LT. 99.9) THEN
           NLAM = 2
           IRMAG = IRM11
           IJ=IJ+1
          ELSE
           IF ( IRM20 .LT. 99.9) THEN
           NLAM = 3
           IRMAG= IRM20
           IJ=IJ+10
          ELSE
           IF ( IRM27 .LT. 99.9) THEN
           NLAM = 4
           IRMAG= IRM27
           IJ=IJ+20
          ELSE
           WRITE(*,'(A)') '  ALL IR WAVELENGTHS WERE 99 OR 0'
           END IF
          END IF
         END IF
```

```
            END IF
C       This subroutine compares visual magnitude for match

             CALL MAG(NLAM,VMAG,IRMAG,T,SRAD,MATCH,NEND,DEND,J,
     +                SMAG,HPEAK)

C       If match = 0 thenmatch is established

            IF ( MATCH .EQ. 0 ) THEN
               NCOUNT=NCOUNT+1

C        This subroutine computes spectral irradiance between two
C           given wave bands....4 and 12

C 76            CALL RADIANCE(T,SRAD,TRAD)

C        Finally, all star match data is written to a file

                  WRITE(NEND,20) NCOUNT,SRA(J),SDEC(J),T,SPECT(J),
     +             SUBSP(J),VMAG(J),SRAD,NLAM,IRMAG,SMAG,HPEAK
 20              FORMAT(1X,I6,1X,F10.5,1X,F10.5,1X,F10.1,1X,A3,1X,
     +               A1,1X,F4.1,E12.5,I2,F4.1,F4.1,2X,E12.5)


 76              CALL RADIANCE(T,SRAD,TRAD)
C              ENDIF
            IF (TRAD .GT. .730E-15) THEN
            WRITE(6,20) NCOUNT,SRA(J),SDEC(J),T,SPECT(J),SUBSP(J),
     +                  VMAG(J),TRAD,NLAM,IRMAG,SMAG,HPEAK
          END IF
          END IF
           WRITE(*,*)NCOUNT
 44        CONTINUE
 99        CONTINUE
 777       CONTINUE
           STOP
           END
C*********************************************************************

       SUBROUTINE ANGLE(SRA,SDEC,VMAG,VMA,SPECT,SUBSP,ICNT)

C   This subroutine compares the right ascension and declination
C   of stars

       COMMON/NUM/ NSAO,NPRI,NSUP,NOUT,NBAND
       COMMON/TREE/JCOUNT
       DIMENSION SRA(30000),SDEC(30000),VMAG(30000)
       COMMON/GL/ RA,DEC,IRM4,EM4,IRM11,EM11,IRM20,EM20,IRM27,EM27
       REAL*4 IRM4,IRM11,IRM20,IRM27
       CHARACTER*2 EM4,EM11,EM20,EM27
       CHARACTER*1 SPECT(30000),SUBSP(30000),VMA(30000)

C   This sets the error bands

           ERR=0.01
           ERAMIN=RA-ERR
           ERAMAX=RA+ERR
           EDCMIN=DEC-ERR
           EDCMAX=DEC+ERR
```

```
C    This initializes ICOUNT to count the star matches

       ICOUNT=0

C    The following loop counts the stars and outputs them to a temporary
C    file to be read in again later

       DO 299 I=1,ICNT

C    Comparing Right Ascension and Declination

          IF (SRA(I) .GE. ERAMIN .AND. SRA(I) .LE. ERAMAX) THEN
           IF (SDEC(I) .GE. EDCMIN .AND. SDEC(I) .LE. EDCMAX) THEN
            ICOUNT=ICOUNT+1
            JCOUNT=JCOUNT+1
            WRITE(NOUT,30)ICOUNT,JCOUNT,SRA(I),RA,SDEC(I),DEC,
     &        VMAG(I),VMA(I),SPECT(I),SUBSP(I),IRM4,EM4,IRM11,
     &        EM11,IRM20,EM20,IRM27,EM27
 30         FORMAT(I6,1X,I6,1X,F10.5,1X,F10.5,F11.5,F10.5,F4.1,1X,
     &        A1,A1,A1,1X,F4.1,1X,A2,1X,F4.1,1X,A2,1X,F4.1,1X,A2,
     &        1X,F4.1,1X,A2)
             ENDIF
          ENDIF
 299       CONTINUE
C           CLOSE(NOUT)
 99        RETURN
           END
C*************************************************************
          SUBROUTINE TEMPER(SPECT,SUBSP,TEMP,ICNT)

C     Spectral Class and Subclass (SPECT and SUBSP) characters are read
C     from the SAO Catalog. The classes correspond to Temperatures. The
C     temperatures below need to be multiplied by 1000 deg K.

          CHARACTER*1 CLASS(11),SBCLASS(11),SCLASS(6),SPECT(30000),
     +                SUBSP(30000)
          REAL*4 TEMP
          REAL TEMPN(11,11),TEMPO(6),TEMPMN(6)
          DATA CLASS /'O','B','A','F','G','K','M','N','S','R','C'/
C          DATA CLASS /'O','B','A','F','G','K','M','R','C','N','S'/
          DATA SBCLASS /'0','1','2','3','4','5','6','7','8','9',' '/
          DATA SCLASS /'A','B','C','D','E','F'/

C      This table plots temperatures with class verses subclass:

         DATA TEMPN/50000.,22000.,10800.,7400.,5900.,5000.,3600.,
     +        5200.,5200.,3600.,3600.,
     +        45000.,20000.,10000.,7200.,5800.,4800.,3400.,5000.,
     +        5000.,3400.,3400.,
     +        45000.,18000.,9800.,7000.,5700.,4600.,3200.,4900.,
     +        4900.,3200.,3200.,
     +        40000.,16000.,9400.,6800.,5600.,4500.,3100.,4700.,
     +        4700.,3100.,3100.,
     +        36000.,15000.,8900.,6700.,5600.,4300.,2900.,4200.,
     +        3900.,2900.,2900.,
     +        36000.,14000.,8500.,6500.,5500.,4200.,2700.,3900.,
     +        3700.,2700.,2700.,
     +        33000.,13000.,8300.,6400.,5400.,4000.,2600.,3700.,
     +        3300.,2600.,2600.,
```

```
     +      28000.,12500.,8000.,6200.,5300.,3900.,2400.,3500.,
     +      3100.,2400.,2400.,
     +      26000.,11700.,7800.,6100.,5200.,3800.,2300.,3200.,
     +      3000.,2300.,2300.,
     +      24000.,113000 ,7600.,6000.,5100.,3700.,2100.,3100.,
     +      2500.,2100.,2100.,
     +      28000.,20000.,10800.,7400 ,5900.,4800.,3200.,5100.,
     +      5100.,3200.,3200./

C      This is extra table information for a higher subclass

      DATA TEMPMN/ 3200.,2900.,2700.,2600.,2400..2400.0/

C      This routine finds temperature based on spectral type

        DO 100 I=1,11
         IF (CLASS(I) .EQ. SPECT(ICNT)) GO TO 11
 100     CONTINUE
         GO TO 56
  11    CONTINUE
        DO 200 J=1,11
          IF (SBCLASS(J) .EQ. SUBSP(ICNT)) GO TO 22
 200    CONTINUE
        DO 300 K=1,6
          IF (SCLASS(K) .EQ. SUBSP(ICNT)) GO TO 33
 300    CONTINUE
          GO TO 56
  22    CONTINUE
C        TEMP= TEMPN(J,I)
        TEMP= TEMPN(I,J)
        GO TO 55

  33    CONTINUE
          TEMP= TEMPMN(K)
        GO TO 55
  44    CONTINUE
  56    TEMP=0.0
  55    CONTINUE
        RETURN
        END

C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
        SUBROUTINE MAG(NLAM,VMAG,IRMAG,TEMP,SRAD,MATCH,NEND,
     +                 DEND,ICNT,SMAG,HPEAK,XX)

C    Using effective temperatures to calculate peak irradiance
C    magnitudes.

        INTEGER NLAM, MATCH, NEND
        CHARACTER*20 DEND
        REAL*4 LAMBDA(4), LAMB
        DIMENSION VMAG(30000)
        REAL*4 HPEAK,RATIO,RAD(16),TEMP,TMAX,DELT
        REAL*4 C1,C2,C3,C4,Q,H0,MAGMAX,MAGMIN,IRMAG
        DATA C1,C2,C3,C4 / 21.199, 14387.9, 3.743E+08, 2.9085E+19/
        DATA SIGMA / 5.669E-12/
        DATA LAMBDA/ 4.2,11.0,20.0,27.0/
        DATA H04,H011,H020,H027/ 3.6E-15,8.7E-17,8.2E-18,2.5E-18/
        DATA RAD /1.2E-10,1.0E-10,7.3E-11,6.0E-11,4.0E-11,3.0E-11,
     .            2.2E-11,1.6E-11,1.0E-11,7.0E-12,4.0E-12,2.0E-12,
```

```
     .            4.1E-12,1.2E-11,6.0E-11,6.0E-11/
C       COMMON/ERR/XX

C      This pairs the wavelength 4.2, 11, 20, or 27 with appropriate
C         spetral irradiance number (Table 4)

       IF( NLAM .EQ. 1)THEN
         LAMB = LAMBDA(1)
         H0= H04
       ELSE
         IF( NLAM .EQ. 2) THEN
           LAMB = LAMBDA(2)
           H0=H011
         ENDIF
         IF( NLAM .EQ. 3) THEN
           LAMB = LAMBDA(3)
           H0=H020
         ENDIF
         IF( NLAM .EQ. 4) THEN
           LAMB = LAMBDA(4)
           H0=H027
         ENDIF
       ENDIF

C    Omitting temperatures and visual magnitudes which are not
C      matchable

        IF ( TEMP .LT. 2000. .OR. TEMP .GT. 28000.) GO TO 99
        IF ( VMAG(ICNT) .EQ. 0.) GO TO 99
         ICOUNT=1
         TMAX=26000.
         DELT=2000.

  10    CONTINUE
        IF (TMAX .LT. 4000.) GO TO 20
        IF (TEMP .GT. TMAX ) GO TO 40
         TMAX= TMAX - DELT
         ICOUNT=ICOUNT + 1
        GO TO 10

  20    CONTINUE
        TMAX= 3000.
        DELT= 1000.
        ICOUNT=13

  30    CONTINUE
        IF (TEMP .GT. TMAX) GO TO 40
          TMAX=TMAX-DELT
          ICOUNT=ICOUNT+1
        GO TO 30

  40    CONTINUE
         RATIO = (TEMP - TMAX)/DELT
         H0PEAK=((RAD(ICOUNT)-RAD(ICOUNT+1))*RATIO)+RAD(ICOUNT+1)

C      Solving for Spectral Irradiance

         HPEAK= H0PEAK * (10.**(-VMAG(ICNT)/2.5))
         Q = C2 / (LAMB * TEMP)
        YO=EXP(Q)
```

```
          HINT = (HPEAK * (Q**5))/(C1 * (EXP(Q)-1))

C    This is the SAO Magnitude of the star at a wavelength that
C    corresponds with the AFGL star

          IF(HINT/H0.LE.0)GO TO 99

C     Checking to see if SMAG is near IRMAG
C       IRMAG is the AFGL IR magnitude 4.2,11.0,20.0,27.0

          MAGMAX=IRMAG+2.
          MAGMIN=IRMAG-2.
          IF (SMAG .GT. MAGMAX .OR. SMAG .LT. MAGMIN) GO TO 99

C      !!!!!!!!!!!!!!!! We have a match !!!!!!!!!!!!!!!!!!!!
C      Calculating the Spectral Irradiance over all wavelengths

          SUM= SIGMA * (TEMP**4)
          SF= (C4*10000.*HPEAK)/((TEMP**5)*C3)
           SRAD=SF*SUM
           MATCH=0

           GO TO 100
 99        CONTINUE
           MATCH=1
 100       CONTINUE
           RETURN
           END

*************************************************************************
          SUBROUTINE RADIANCE(T,SRAD,TRAD)

C       This subroutine converts spectral irradiance into readings
C         in appropriate wavelengths :between 4 and 12


          REAL*4 TRAD,LAM,T4,T12,LAM4A,LAM12A,P4,P12,MMK(93)
          REAL*4 ELAM(93)
          INTEGER ICNT

          DATA MMK/555.6,666.7,777.7,888.9,1000.0,1111.1,1222.2,
     +           1333.3,1444.4,1555.6,1666.7,1777.8,1888.9,2000.,
     +           2111.1,2222.2,2333.3,2444.4,2555.6,2666.7,2777.8,
     +           2888.9,3000.0,3111.1,3222.2,3333.3,3444.4,3555.5,
     +           3666.6,3777.7,3888.8,4000.0,4111.1,4222.2,4333.3,
     +           4444.4,4555.6,4666.7,4777.8,4888.9,5000.0,5111.1,
     +           5222.2,5333.3,5444.4,5555.6,5666.7,5777.8,5888.9,
     +           6000.0,6111.1,6222.2,6333.3,6444.4,6555.6,6666.7,
     +           6777.8,6888.9,7000.0,7111.1,7222.2,7333.3,7444.4,
     +           7555.6,7666.7,7777.8,7888.9,8000.0,8111.1,8222.2,
     +           8333.3,8888.9,9444.4,10000.,10555.6,11111.1,11666.7,
     +           12222.2,12777.8,13333.3,13888.9,14444.4,15000.0,
     +           15555.6,16111.1,16666.7,22222.2,27777.8,33333.3,
     +           38888.9,44444.4,50000.0,55555.6/

          DATA ELAM/.0000000170,.000000756,.0000106,.0000738,.000321,
     +           .00101,.00252,.00531,.00983,.01643,.02537,.03677,
     +           .05059,.06672,.08496,.10503,.12665,.14953,.17337,
     +           .19789,.22285,.24803,.27322,.29825,.32300,.34734,
     +           .37118,.39445,.41708,.43905,.46031,.48085,.50066,
```

```
     +          .51974,.53809,.55573,.57267,.58891,.60449,.61941,
     +          .63371,.64740,.66051,.67305,.68506,.69655,.70754,
     +          .71806,.72813,.73777,.74700,.75583,.76429,.77238,
     +          .78014,.78757,.79469,.80152,.80806,.81433,.82035,
     +          .82612,.83166,.83698,.84209,.84699,.85171,.85624,
     +          .86059,.86477,.86880,.88677,.90168,.91414,.92462,
     +          .93349,.94104,.94751,.95307,.95778,.96207,.96572,
     +          .96892,.97174,.97423,.97644,.98915,.99414,.99649,
     +          .99773,.99845,.99889,.99918/

C     Converting temperatures to appropriate wavebands

          T4 = T*4
          T12= T*12

C     Interpolating for 4 microns

          DO 44 ICNT=1,93
            IF(MMK(ICNT).EQ.T4)THEN
              LAM4A=ELAM(ICNT)
              GO TO 33
            ENDIF
            IF(MMK(ICNT).GT.T4)THEN
              P4=((ELAM(ICNT)-ELAM(ICNT-1))*(MMK(ICNT)-T4)/
     +          (MMK(ICNT)-MMK(ICNT-1)))
              LAM4A=ELAM(ICNT)-P4
              GO TO 33
            ENDIF
 44         CONTINUE

C      Interpolating for 12 microns

 33         DO 22 JCNT=1,93
              IF(T12.GT.MMK(93))THEN
                LAM12A=1.
                GO TO 12
              ENDIF
              IF(MMK(JCNT).EQ.T12)THEN
                LAM12A=ELAM(JCNT)
                GO TO 11
              ENDIF
              IF(MMK(JCNT).GT.T12)THEN
                P12=((ELAM(JCNT)-ELAM(JCNT-1))*(MMK(JCNT)-T12)/
     +            (MMK(JCNT)-MMK(JCNT-1)))
                LAM12A=ELAM(JCNT)-P12
                GO TO 12
              ENDIF

 22         CONTINUE
 12         CONTINUE

C      This sights the difference in the two interpolated wavelengths
C         and finds the appropriate spectral irradiance

            LAM=LAM12A-LAM4A
            TRAD= LAM*SRAD

 11          CONTINUE
            RETURN
            END
```

```
C  This allows the SAO map to be read in by array of up to 30000 elements

      CHARACTER*1 SPECT,SUBSP,VMA
      INTEGER SAON,FILECNT,XX
      CHARACTER*20 SAO,PRI,SUP,OUT,BAND,DEND
      REAL*4 VMAG,IRMAG,TEMP,IRM4,IRM11,IRM20,IRM27,PM,TRAD
      CHARACTER*2 EM4,EM11,EM20,EM27,ANS
      NCOUNT=0
C   This defines the files which are opened

      DO 777 FILECNT=1,18
       IF(FILECNT.EQ.1)THEN
         SAO= '[STARS]SAO85.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD85.DAT'
         GO TO 1
       ENDIF
        IF(FILECNT.EQ.2)THEN
         SAO= '[STARS]SAO75.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD75.DAT'
         GO TO 1
       ENDIF
        IF(FILECNT.EQ.3)THEN
         SAO= '[STARS]SAO65.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD65.DAT'
         GO TO 1
       ENDIF
        IF(FILECNT.EQ.4)THEN
         SAO= '[STARS]SAO55.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD55.DAT'
         GO TO 1
       ENDIF
        IF(FILECNT.EQ.5)THEN
         SAO= '[STARS]SAO45.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD45.DAT'
         GO TO 1
       ENDIF
        IF(FILECNT.EQ.6)THEN
         SAO= '[STARS]SAO35.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD35.DAT'
         GO TO 1
       ENDIF
        IF(FILECNT.EQ.7)THEN
         SAO= '[STARS]SAO25.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD25.DAT'
         GO TO 1
       ENDIF
        IF(FILECNT.EQ.8)THEN
         SAO= '[STARS]SAO15.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD15.DAT'
         GO TO 1
       ENDIF
        IF(FILECNT.EQ.9)THEN
         SAO= '[STARS]SAO05.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD05.DAT'
         GO TO 1
        ENDIF
        IF(FILECNT.EQ.10)THEN
         SAO= '[STARS]SAO-05.DAT'
         DEND='[STERRETT.STARSTUFF]SAORAD-05.DAT'
         GO TO 1
```

```
          ENDIF
           IF(FILECNT.EQ.11)THEN
            SAO= '[STARS]SAO-15.DAT'
            DEND='[STERRETT.STARSTUFF]SAORAD-15.DAT'
            GO TO 1
          ENDIF
           IF(FILECNT.EQ.12)THEN
            SAO= '[STARS]SAO-25.DAT'
            DEND='[STERRETT.STARSTUFF]SAORAD-25.DAT'
            GO TO 1
          ENDIF
           IF(FILECNT.EQ.13)THEN
            SAO= '[STARS]SAO-35.DAT'
            DEND='[STERRETT.STARSTUFF]SAORAD-35.DAT'
            GO TO 1
          ENDIF
           IF(FILECNT.EQ.14)THEN
            SAO= '[STARS]SAO-45.DAT'
            DEND='[STERRETT.STARSTUFF]SAORAD-45.DAT'
            GO TO 1
          ENDIF
           IF(FILECNT.EQ.15)THEN
            SAO= '[STARS]SAO-55.DAT'
            DEND='[STERRETT.STARSTUFF]SAORAD-55.DAT'
            GO TO 1
          ENDIF
           IF(FILECNT.EQ.16)THEN
            SAO= '[STARS]SAO-65.DAT'
            DEND='[STERRETT.STARSTUFF]SAORAD-65.DAT'
            GO TO 1
          ENDIF
           IF(FILECNT.EQ.17)THEN
            SAO= '[STARS]SAO-75.DAT'
            DEND='[STERRETT.STARSTUFF]SAORAD-75.DAT'
            GO TO 1
          ENDIF
           IF(FILECNT.EQ.18)THEN
            SAO= '[STARS]SAO-85.DAT'
            DEND='[STERRETT.STARSTUFF]SAORAD-85.DAT'
           END IF
C   Defining File Numbers

          DATA NSAO,NPRI,NSUP,NOUT,NEND /10,20,30,40,50/
 1        OPEN(NSAO,FILE=SAO,STATUS='OLD',READONLY)
          OPEN(NEND,FILE=DEND,STATUS='NEW')

C   Defining constants

          DATA PI, C1, C2 /3.141592654, 3.7415E+04, 1.43879E+04/

C   This loop reads in the SAO map in an array for comparison

          WRITE(*,*)' READING IN SAO MAP...',FILECNT
          DO 299 II=1,27000

 25        FORMAT(I7,A1,F10.5,A1,F11.5,1X,F4.1,1X,F4.1,A1,A1,1X,A1,
     +      A1,A1,A2,A1,F10.8,F11.8)
           READ(NSAO,25,END=299)SAON,B,SRA,SND,SDEC,PM,VMAG,
     +      SPECT,SUBSP,DV,VMA,PMA,CI1,CI2,SRAR,SDECR
          ICNT=II
```

```
C   Calculating the actual temperature of the star from the spectral
C   class and subclass

            CALL TEMPER(SPECT,SUBSP,T,II)
            IF (T .EQ. 0.) GO TO 44

C      This subroutine calculates spectral irradiance over
C      all wavelengths.

            CALL MAG(NLAM,VMAG,T,SRAD,MATCH,J,HPEAK)

C      If match = 0 then match is established

           IF ( MATCH .EQ. 1 ) GO TO 44

C       This subroutine computes spectral irradiance between two
C          given wave bands....lambda1 and lambda2

 76           CALL RADIANCE(T,SRAD,SRAD2)

C       Finally, all star data is written to a file

 20            FORMAT(1X,I6,1X,F10.5,1X,F10.5,1X,F10.1,1X,A1,1X,
     +            A1,1X,F4.1,E12.5)

          IF (SRAD2 .GT. 1.0E-15) THEN
          NCOUNT=NCOUNT+1
          WRITE(NEND,20) NCOUNT,SRA,SDEC,T,SPECT,
     *             SUBSP,VMAG,SRAD2
          END IF
 44        CONTINUE
 99        CONTINUE
 299       CONTINUE
 777       CONTINUE
           WRITE(*,*)NCOUNT
           STOP
           END

C*****************************************************************
        SUBROUTINE TEMPER(SPECT,SUBSP,TEMP,ICNT)

C    Spectral Class and Subclass (SPECT and SUBSP) characters are read
C    from the SAO Catalog.  The classes correspond to Temperatures.  The
C    temperatures can be determined from the tables.

        CHARACTER*1 CLASS(11),SBCLASS(11),SCLASS(6),SPECT,
     +              SUBSP
        REAL*4 TEMP
        REAL TEMPN(11,11),TEMPO(6),TEMPMN(6)
        DATA CLASS /'O','B','A','F','G','K','M','N','S','R','C'/
C        DATA CLASS /'O','B','A','F','G','K','M','R','C' 'N','S'/
        DATA SBCLASS /'0','1','2','3','4','5','6','7' '8','9',' '/
        DATA SCLASS /'A','B','C','D','E','F'/

C     This table plots temperatures with class verses subclass:

       DATA TEMPN/50000.,22000.,10800.,7400.,5900.,5000.,3600.,
     +      5200.,5200.,3600.,3600.,
     +      45000.,20000.,10000.,7200.,5800.,4800.,3400.,5000.,
```

```
     +      5000.,3400.,3400.,
     +      45000.,18000.,9800.,7000.,5700.,4600.,3200.,4900.,
     +      4900.,3200.,3200.,
     +      40000.,16000.,9400.,6800.,5600.,4500.,3100.,4700.,
     +      4700.,3100.,3100.,
     +      36000.,15000.,8900.,6700.,5600.,4300.,2900.,4200.,
     +      3900.,2900.,2900.,
     +      36000.,14000.,8500.,6500.,5500.,4200.,2700.,3900.,
     +      3700.,2700.,2700.,
     +      33000.,13000.,8300.,6400.,5400.,4000.,2600.,3700.,
     +      3300.,2600.,2600.,
     +      28000.,12500.,8000.,6200.,5300.,3900.,2400.,3500.,
     +      3100.,2400.,2400.,
     +      26000.,11700.,7800.,6100.,5200.,3800.,2300.,3200.,
     +      3000.,2300.,2300.,
     +      24000.,113000.,7600.,6000.,5100.,3700.,2100.,3100.,
     +      2500.,2100.,2100.,
     +      28000.,20000.,10800.,7400.,5900.,4800.,3200.,5100.,
     +      5100.,3200.,3200./

C       This is extra table information for a higher subclass.

      DATA TEMPMN/ 3200.,2900.,2700.,2600.,2500.,2400.0/

C       This routine finds temperature based on spectral type.

        DO 100 I=1,11
         IF (CLASS(I) .EQ. SPECT) GO TO 11
 100    CONTINUE
        GO TO 56
  11    CONTINUE
        DO 200 J=1,11
          IF (SBCLASS(J) .EQ. SUBSP) GO TO 22
 200    CONTINUE
        DO 300 K=1,6
          IF (SCLASS(K) .EQ. SUBSP) GO TO 33
 300    CONTINUE
          GO TO 56
  22    CONTINUE
        TEMP= TEMPN(I,J)
        GO TO 55

  33    CONTINUE
        TEMP= TEMPMN(K)
        GO TO 55
  44    CONTINUE
  56    TEMP=0.0
  55    CONTINUE
        RETURN
        END

C&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
        SUBROUTINE MAG(NLAM,VMAG,TEMP,SRAD,MATCH,ICNT,
     +     HPEAK)

C    Using effective temperatures to calculate peak irradiance
C    magnitudes.

       INTEGER NLAM, MATCH, NEND
       CHARACTER*20 DEND
```

```
      REAL*4 LAMBDA(4), LAMB
      REAL*4 HPEAK,RATIO,RAD(16),TEMP,TMAX,DELT
      REAL*4 C1,C2,C3,C4,Q,H0,MAGMAX,MAGMIN,IRMAG
      DATA C1,C2,C3,C4 / 21.199, 14387.9, 3.743E+08, 2.9085E+19/
      DATA SIGMA / 5.669E-12/
      DATA LAMBDA/ 4.2,11.0,20.0,27.0/
      DATA H04,H011,H020,H027/ 3.6E-15,8.7E-17,8.2E-18,2.5E-18/
      DATA RAD /1.2E-10,1.0E-10,7.3E-11,6.0E-11,4.0E-11,3.0E-11,
     .          2.2E-11,1.6E-11,1.0E-11,7.0E-12,4.0E-12,2.0E-12,
     .          4.1E-12,1.2E-11,6.0E-11,6.0E-11/
C      COMMON/ERR/XX

C     This pairs the wavelength 4.2, 11, 20, or 27 with appropriate
C        spetral irradiance nu   r (Table 4)

      IF( NLAM .EQ. 1)THEN
        LAMB = LAMBDA(1)
        H0= H04
      ELSE
        IF( NLAM .EQ. 2) THEN
          LAMB = LAMBDA(2)
          H0=H011
        ENDIF
        IF( NLAM .EQ. 3) THEN
          LAMB = LAMBDA(3)
          H0=H020
        ENDIF
        IF( NLAM .EQ. 4) THEN
          LAMB = LAMBDA(4)
          H0=H027
        ENDIF
       ENDIF

C    Omitting temperatures and visual magnitudes which are not
C      matchable

       IF ( TEMP .LT. 2000. .OR. TEMP .GT. 28000.) GO TO 99
       IF ( VMAG .EQ. 0.) GO TO 99
        ICOUNT=1
        TMAX=26000.
        DELT=2000.

  10    CONTINUE
        IF (TMAX .LT. 4000.) GO TO 20
        IF (TEMP .GT. TMAX ) GO TO 40
         TMAX= TMAX - DELT
         ICOUNT=ICOUNT + 1
        GO TO 10

  20    CONTINUE
        TMAX= 3000.
        DELT= 1000.
        ICOUNT=13

  30    CONTINUE
        IF (TEMP .GT. TMAX) GO TO 40
          TMAX=TMAX-DELT
          ICOUNT=ICOUNT+1
        GO TO 30
```

```
 40      CONTINUE
         RATIO = (TEMP - TMAX)/DELT
         H0PEAK=((RAD(ICOUNT)-RAD(ICOUNT+1))*RATIO)+RAD(ICOUNT+1)

C       Solving for Spectral Irradiance

         HPEAK= H0PEAK * (10.**(-VMAG/2.5))

C      Calculating the Spectral Irradiance over all wavelengths

         SUM= SIGMA * (TEMP**4)
         SF= (C4*10000.*HPEAK)/((TEMP**5)*C3)
          SRAD=SF*SUM
          MATCH=0

          GO TO 100
 99       CONTINUE
          MATCH=1
 100      CONTINUE
          RETURN
          END

*****************************************************************************
         SUBROUTINE RADIANCE(T,SRAD,SRAD2)

C       This subroutine converts spectral irradiance for all wavelengths
C       to the appropriate spectral irradiance between lambda1 and lambda2


         REAL*4 TRAD,LAM,T4,T12,LAM4A,LAM12A,P4,P12,MMK(93)
         REAL*4 ELAM(93)
         INTEGER ICNT

         DATA MMK/555.6,666.7,777.7,888.9,1000.0,1111.1,1222.2,
     +           1333.3,1444.4,1555.6,1666.7,1777.8,1888.9,2000.,
     +           2111.1,2222.2,2333.3,2444.4,2555.6,2666.7,2777.8,
     +           2888.9,3000.0,3111.1,3222.2,3333.3,3444.4,3555.5,
     +           3666.6,3777.7,3888.8,4000.0,4111.1,4222.2,4333.3,
     +           4444.4,4555.6,4666.7,4777.8,4888.9,5000.0,5111.1,
     +           5222.2,5333.3,5444.4,5555.6,5666.7,5777.8,5888.9,
     +           6000.0,6111.1,6222.2,6333.3,6444.4,6555.6,6666.7,
     +           6777.8,6888.9,7000.0,7111.1,7222.2,7333.3,7444.4,
     +           7555.6,7666.7,7777.8,7888.9,8000.0,8111.1,8222.2,
     +           8333.3,8888.9,9444.4,10000.,10555.6,11111.1,11666.7,
     +           12222.2,12777.8,13333.3,13888.9,14444.4,15000.0,
     +           15555.6,16111.1,16666.7,22222.2,27777.8,33333.3,
     +           38888.9,44444.4,50000.0,55555.6/

         DATA ELAM/.0000000170,.000000756,.0000106,.0000738,.000321,
     +           .00101,.00252,.00531,.00983,.01643,.02537,.03677,
     +           .05059,.06672,.08496,.10503,.12665,.14953,.17337,
     +           .19789,.22285,.24807,.27322,.29825,.32300,.34734,
     +           .37118,.39445,.41700,.43905,.46031,.48085,.50066,
     +           .51974,.53809,.55573,.57267,.58891,.60449,.61941,
     +           .63371,.64740,.66051,.67305,.68506,.69655,.70754,
     +           .71806,.72813,.73777,.74700,.75583,.76429,.77238,
     +           .78014,.78757,.79469,.80152,.80806,.81433,.82035,
     +           .82612,.83166,.83698,.84209,.84699,.85171,.85624,
     +           .86059,.86477,.86880,.88677,.90168,.91414,.92462,
     +           .93349,.94104,.94751,.95307,.95778,.96207,.96572,
```

```
     +          .96892,.97174,.97423,.97644,.98915,.99414,.99649,
     +          .99773,.99845,.99889,.99918/

C     Converting temperatures to appropriate wavebands

          T1=T*.5
          T2=T*.9

C     Interpolating for lambda1

          DO 44 ICNT=1,93
            IF(MMK(ICNT).EQ.T1)THEN
              LAM4A=ELAM(ICNT)
              GO TO 33
            ENDIF
            IF(MMK(ICNT).GT.T1)THEN
              P4=((ELAM(ICNT)-ELAM(ICNT-1))*(MMK(ICNT)-T1)/
     +          (MMK(ICNT)-MMK(ICNT-1)))
              LAM4A=ELAM(ICNT)-P4
              GO TO 33
            ENDIF
 44        CONTINUE

C      Interpolating for lambda2

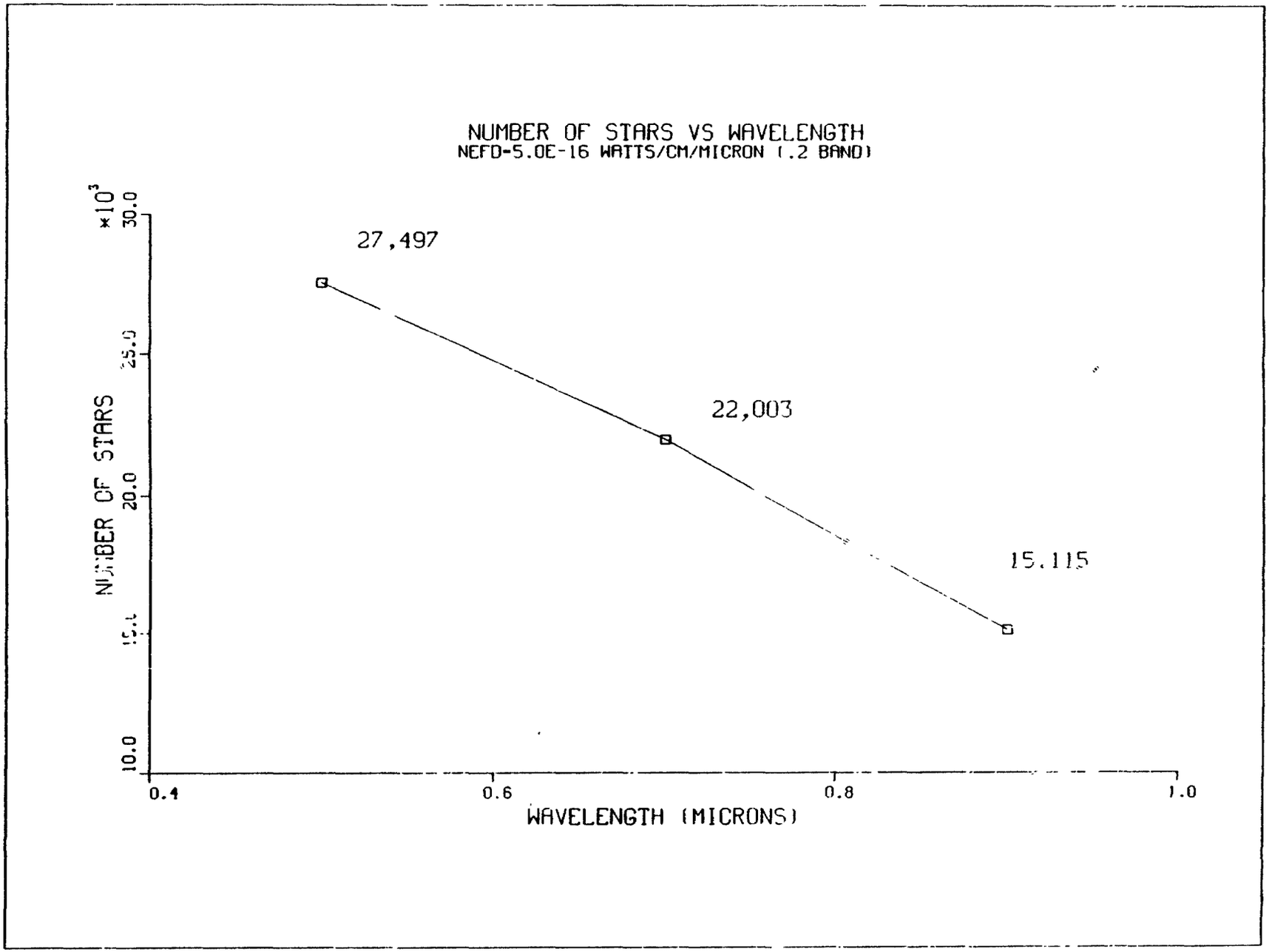
 33         DO 22 JCNT=1,93
              IF(T12.GT.MMK(93))THEN
                LAM12A=1.
                GO TO 12
              ENDIF
              IF(MMK(JCNT).EQ.T2)THEN
                LAM12A=ELAM(JCNT)
                GO TO 11
              ENDIF
              IF(MMK(JCNT).GT.T2)THEN
                P12=((ELAM(JCNT)-ELAM(JCNT-1))*(MMK(JCNT)-T2)/
     +            (MMK(JCNT)-MMK(JCNT-1)))
                LAM12A=ELAM(JCNT)-P12
                GO TO 12
              ENDIF

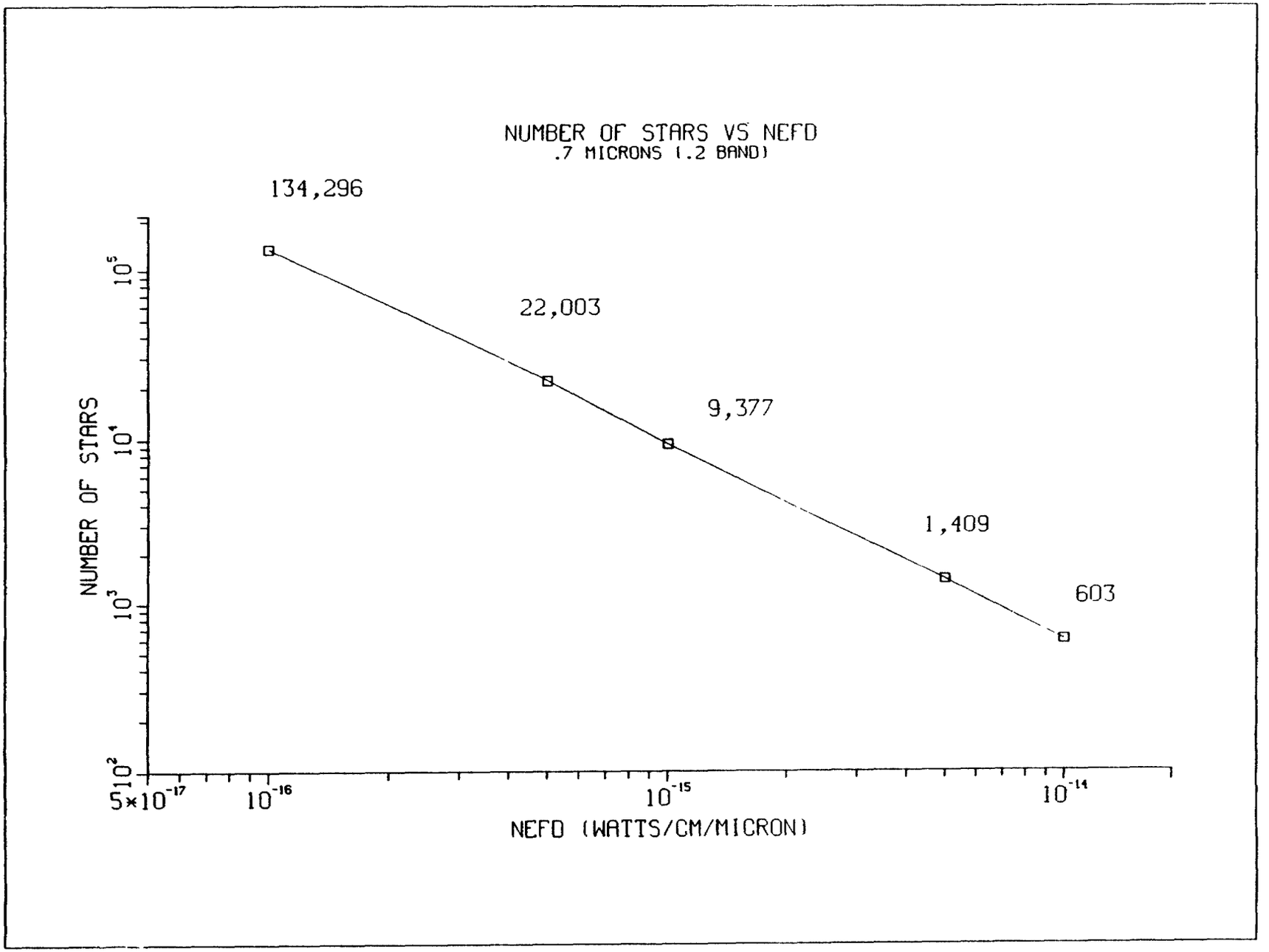
 22         CONTINUE
 12         CONTINUE

C      This sights the difference in the two interpolated wavelengths
C         and finds the appropriate spectral irradiance

           LAM=LAM12A-LAM4A
           SRAD2=LAM*SRAD

 11         CONTINUE
           RETURN
           END
```


NUMBER OF STARS VS WAVELENGTH
NEFD-5.0E-16 WATTS/CM/MICRON (.2 BAND)
NUMBER OF STARS
×10³
30.0
20.0
10.0
27,497
22,003
15,115
0.4
0.6
0.8
1.0
WAVELENGTH (MICRONS)


NUMBER OF STARS VS NEFD
.7 MICRONS (.2 BAND)
134,296
22,003
9,377
1,409
603
NUMBER OF STARS
10^5
10^4
10^3
5×10^2
5×10^{-17}
10^{-16}
10^{-15}
10^{-14}
NEFD (WATTS/CM/MICRON)

TERMS

AFGL - Air Force Geophysics Lab

Alpha - program written by Nicki Cook to organize the 500 brightest stars alphabetically by constellation and descending magnitude.

Beta - program written by Nicki Cook to organize the Alpha data files according to constellation groupings and descending visual magnitude

declination - the angular distance of a celestial body north or south from the celestial equator

D.U.M.B.O. - program (based on Rockwell International algorithms) written by Laura Sterrett and modified by Tim Poth and Nicki Cook to compare stars on the SAO and RAFGL star maps, check for matches in position and visual magnitude, compute total spectral irradiance, and find the spectral irradiance in a particular waveband

HSAP - High School Apprenticeship Program

N.E.F.D. - Noise Equivalent Flux Density

right ascension - the angular distance of the hour circle of a celestial body from the vernal equinox, measured eastward along the celestial equator and expressed in degrees (from 0 to 360) or, more commonly, in hours (from 0 to 24), minutes, and seconds

SAI - Guided Interceptor Branch in the Strategic

Analysis division

SAIC - Science Applications International Corporation

SETA - Scientific and Engineering Technical Assistant

uranographic - dealing with the branch of astronomy concerned with the description of the heavens and the mapping of the stars

SAO - Smithsonian Astrophysical Observatory

S.A.S.S.E. - the title of Nicki Cook's project for the 1990 summer HSAP; it stands for "Star Availability for Sensor - Specific Evaluation

spectral irradiance - total energy emitted in all wavelengths (usually light energy)

spectral type - system invented by Anne Cannon at Harvard which uses letter/number pairs to identify the temperature and color of a star

star addresses - phrase coined which refers to the identifying names and symbols of a star, specifically: constellation name, Greek letter, and star name

visual magnitude - brightness judged by the response of the human eye

BIBLIOGRAPHY

Bueche, F., Principles of Physics of Principles, Edition III, McGraw-Hill Book Com., NY, 1977

Delporte, E., Atlas du Ciel, Paris, France, 1955

Flammarion, Camille, The Flammarion Book of Astronomy, Simon and Schuster, NY, 1964

Holman, J.P., Heat Transfer, McGraw-Hill Book Com., USA, 1981

Metcalf, Michael, Effective FORTRAN 77, Clarendon Press, Oxford, 1985

Neufield, Victoria, Webster's New World Dictionary, Edition III, NY, 1988

Wyatt, Stanley P., Principles of Astronomy, Edition III, Allyn and Bacon Inc., Boston, MA, 1977

Heather Cox, HSAP
Mentor : Michael E Nixon

Development of a Customized Database System for the Distribution of EPIC Hydrocode Software

August 15, 1990

I. Introduction

This summer was the first and last summer I worked in the High School Apprenticeship Program I worked in the Warheads B Branch of the Munitions Division of Eglin AFATL, or MNW for short. My summer job was invaluable to me as a learning experience in both computer sciences and life in general. My mentor, Michael E Nixon, and my other coworkers taught me more this summer than all my years in high school about working in a professional environment and preparing for life in the "Real World."

II. Acknowledgements

I would like to thank all the people who helped me this summer with my project and my other jobs. My fellow apprentices all made the days more enjoyable, especially Ken Gage, my coworker who "showed me the ropes" at MNW. I'd also like to thank Lori Perillo and Bizhan Aref for taking care of Ken and me while our mentor was out of town. Bill Cook, Capt. Rich Guba, Al Weimorts, and David Wagnon all graciously loaned me their workstations during the summer so that I could write and run all my programs. My constant sources of personal computing advice were Darren Boisjolie and Capt. Ernie Staubs. I'd also like to thank those who are in charge of the HSAP program for giving me the chance to participate in such a wonderful summer experience. But most of all I thank my mentor, Michael Nixon, for giving me a great summer and lots of good advice about life, the universe, and everything.

III. Munitions-Warheads Division (MNW)

MNW is the warheads branch of the munitions division. It is composed of several smaller teams. The first is the lethal mechanics team, which makes sure a warhead kills its target. The warhead mechanics team develops the mechanical aspects of a bomb. The AWEF (Advanced Weapons Evaluation Facility) will test warheads of heavy metals, such as depleted uranium, once it is opened and put into action. The last team of MNW is the computational mechanics team, in which I worked this summer. The job of this team is to run computer simulations of warhead impacts on targets, which are then compared to the actual tests run by the other MNW teams.

IV. Background of Research Project

My mentor is planning to leave Eglin temporarily at the end of this summer to attend graduate school at the University of Florida. My project was to prepare those taking over his duties to do so easily and quickly. One of Mr Nixon's duties at Eglin is the management of the distribution of EPIC Hydrocode software, and my job was to create a databse program to make it easier for someone to access and edit the distribution files of EPIC. The results of my summer work can be seen in the documentation of the program that I wrote, which is included in the following pages of this report.

V. Documentation and Results

Database File System Documentation

The purpose of this dBase program is to organize and allow easy access to the contractor, university, and government files of EPIC Hydrocode distribution. This program is user friendly, and the menus will help you through the program, but this documentaion also contains some things you need to know.

Use of the File System

The first step in using this program is loading it. You may need to transfer tne files for this program from a floppy disk onto the hard drive of the workstation you plan to use. Place the floppy in the "A" drive, and use the "COPY" command to transfer the files into the dBase directory. (Type "cd dbase (return)" to change to the dbase directory).

Example: copy A:epicc.* C:epicc.* (return)

The names of the files you need to copy are:

epicc.* name.*

epicg.* name2.*

epicu.* name3.*

dbepic.*

After copying these files type "dbase (return)" and you'll be in dBase.

Once you get into dBase, you need to get to the dot prompt. Press the return key to get to the assist menu, then Esc to the dot prompt. You can begin the program here by typing

"do dbepic (return)"

Once the program begins, you will see the mair menu of the program, like the one below:

EPIC Distribution File Manager

1.Access Contractor Files
2.Access University Files
3.Access Government Files
4.Terminate Session

ENTER OPTION:

Options 1-3 will show you new menus, while option 4 will end the program and take you back out of dBase. Just press the key which corresponds with your choice of options, and you'll see the new menu. (NOTE: If you press a key which is not one of the options, the dot prompt will return. Just type "do dbepic (return)" and the program will start over. Do this any time the program returns you to the dot prompt.)

The menus for contractor, government, and university files are all basically the same. An example of the contractor menu can be seen below.

Contractor File Manager

1.Edit Files
2.Append Files
3.Print Files
4.Search by Name
5.Return to Main Menu

ENTER OPTION:

Explanation of Options:

1.Edit Files: When you choose this option, the program shows you the first company's record. You can edit this record by typing over words you want to change, or you can insert words by turning on insert mode, by pressing the Ins key. The symbol "Ins" will appear at the top of the screen when you're in insert mode. The keys on the keyboard work like an normal editor-use the cursor keys or the return key to move around in the record, and to move from record to record use the PgUp and PgDn keys (you need to hit the PgDn twice to move to the next record, but to move to the previous record hit the PgUp once). When you want to return to the menu, hit the Esc key.

2. Append Files: This option allows you to add new records to the end of the file. It automatically positions you on a blank record after the last existing record. Just type in your information into the record, and it is automatically saved when you return to the menu. If you want to add more than one record, type in the first, then hit PgDn twice and you will be given another blank record. Just do this until you are finished, then hit Esc to return to the menu.

3. Print Files: This command automatically prints every record in the file you're using. The files will not be shown on the screen, they will only be sent to the printer. Use this option only if you want to print a copy of all the records in a file. To print one record at a time, use option 1,2,

or 4 to get the record you want to print on the screen, then do a screen dump by typing "(shift)(PrScr)."

4. Search: The Search option allows you to edit a specific record without having to scroll through all the other records in a file. When you choose this option you will be presented with a place to type in the name of the company, university, or government installation you wish to see. (You must type in the exact name that's already on the record, or the search will be unsuccessful, and you'll be returned to the menu.)

5. Return to Main Menu: This option is self-explanatory; it returns you to the main menu, where you can go to another file or end the program and exit dBase.

Ending the Session

You'll need to save all the changes in information that you have made onto the floppy disk. After ending the program, place the disk in the A drive, and execute the following commands:

"copy c:epicc.dbf a:epicc.dbf"

"copy c:epicg.dbf a:epicg.dbf"

"copy c:epicu.dbf a:epicu.dbf"

Make sure the files were copied by changing to the A drive (type "A: (return)"), and doing a directory, "dir (return)", and checking for those files in the list. If they were not correctly copied, do the "copy" over until it works.

If the hard drive on the workstation you are using needs to be cleaned up after your session, just use the "del" command to remove all the files you put onto the disk at the beginning of the session.

How It Works

The file system consists of one main program (which creates the main menu) and three subprograms (one for contractors, one for universities, and one for government agencies). Each of the three subprograms has its own subprogram, which allows it to perform the search option. Hard copies of these programs can be seen on the following pages, and they are each followed by an explanation of the commands in that program.

DBEPIC.PRG

```
set talk off                                          1
set bell off
set escape off
set status off
choice=" "                                            5

do while .t.

  choice=" "
  @ 3,3 to 14,46 double                              10
  @ 5,10 say "EPIC Distribution File Manager"
  @ 7,12 say "1. Access Contractor Files"
  @ 8,12 say "2. Access University Files"
  @ 9,12 say "3. Access Government Files"
  @ 10,12 say "4. Terminate Session"                 15
  @ 12,15 say "ENTER OPTION:"
  @ 12,29 get choice
  read
```

```
do case                                                    20

   case choice = "1"
    do epicc
    close all
    clear all                                              25
    clear
    loop

   case choice = "2"
    do epicu                                               30
    close all
    clear all
    clear
    loop
                                                           35
   case choice = "3"
    do epicg
    close all
    clear all
    clear                                                  40
    loop

   case choice = "4"
    clear
    @ 13,23 say " Don't forget to save your "              45
    @ 14,23 say "changes on the floppy disk!"
    read
    clear all
    close all
    clear                                                  50
    quit

  end do

end do                                                     55
```

Lines 1-4 of the program DBEPIC (standing for DataBase EPIC), set the environment for the program. The line "set talk off" tells the computer not to print normal status messages on the screen. Line 2, "set bell off", turns off the warning bell. "set escape off" tells the computer to save edits instead of abandoning them as it would normally do when the Esc key is pressed. Line 4, "set status off", turns off the command line

that normally appears at the bottom of the screen to tell you what operational modes you are in.

Line 5 of the program initiates a variable of character type which is one character long. This is for the option you choose from the menu. Line 7 tells the computer to perform the tasks given it as long as the specified conditions are true: "do while .t."

Lies 9-18 set up the first menu you see when the program is run. They tell the computer what to print on which line of the screen. Line 9 once again tells the computer to expect a one character variable, and line 17 tells it to get that variable from the user and proceed to the cases, which correspond with the possible choices. "Read" tells the computer to read the information in the lines above and execute it.

Line 20, "do case" tells the computer that it needs to do one of the following cases, based on the choice the user entered at the menu. Case 1 tells the computer to execute another program, epicc, which allows the user access to the contractor files. "Close all", "clear all", and "clear" tell the computer to finish what it's doing, close everything up, and clear the screen. "Loop" causes the computer to return to the main menu to receive another command from the user. Cases 2 and 3 do the same thing as Case 1, except Case 2 sends the computer to epicu (university files), and Case 3 sends it to epicg (government files).

Case 4, lines 43-51, makes the computer clear the screen, print a message to the user, and then exit dBase. The do cases are closed and the program ends with "end do."

EPICC.PRG

```
set talk off                                          1
set bell off
set escape off
set status off
choice= " "                                           5

do while .t.

  choice = " "
  @ 3,3 to 15,46 double                              10
  @ 5,13 say "Contractor File Manager"
  @ 7,15 say "1. Edit Files"
  @ 8,15 say "2. Append Files"
  @ 9,15 say "3. Print Files"
  @ 10,15 say "4. Search by Name"                    15
  @ 11,15 say "5. Return to Main Menu"
  @ 13,15 say "ENTER OPTION: "
  @ 13,29 get choice
  read                                               20

  do case

  case choice = "1"
     use epicc                                       25
     set format to epicc
     edit
     close all
     clear all
     clear                                           30
     loop

  case choice = "2"
     use epicc
     set format to epicc                             35
     append
     close all
     clear all
     clear
     loop                                            40

  case choice = "3"
     use epicc
```

```
      set device to print
      do while .not. eof()
        @  3,5 SAY "COMPANY NAME:"
        @  3,20 say CONAME
        @  5,5 SAY "ADDRESS:"
        @  5,30 SAY "COMPANY POC:"
        @  5,55 SAY "GOVERNMENT POC:"
        @  6,5 say ADDRESS1
        @  6,30 say CPOC1
        @  6,55 say GPOC1
        @  7,5 say ADDRESS2
        @  7,30 say CPOC2
        @  7,55 say GPOC2
        @  8,5 say ADDRESS3
        @  8,30 say CPOC3
        @  8,55 say GPOC3
        @  9,5 say ADDRESS4
        @  9,30 say CPOC4
        @  9,55 say GPOC4
        @  11,5 SAY "CONTRACT #:"
        @  11,17 say CONTRCTNO
        @  11, 35 SAY "RELEASE FORM:"
        @  11, 49 say REL1
        @  12, 5 SAY "VERSION:"
        @  12, 14 say VERSION
        @  14,5 SAY "CORRESPONDENCE:"
        @   15,5 say COR1
        @  16,5 say COR2
        @  17,5 say COR3
        @  18,5 say COR4
        @  19,5 say COR5
        @  20,5 say COR6
        @  21,5 say COR7
        @  22,5 say COR8
        @  23,5 SAY "NOTES:"
        @  24,5 say NOT1
        skip
      enddo
      set device to screen
      close all
      clear all
      clear
      loop

   case choice = "4"
      clear all
      close all
      clear
      do name

   case choice = "5"
      clear all
      close all
```

```
            clear
            do dbepic
                                                              100
          enddo

        enddo
```

The first few lines of epicc (contractor file manager) are basically the same as dbepic-the program environment is specified, and a variable "choice" is initiated. A "do while" loop is established, and the computer then goes on to the next few lines, which tell it to print a menu on the screen, and accept the value of "choice" from the user. It then goes on to the do cases, using the choice of the user.

Case 1 tells the computer to use the epicc dBase file (epicc.dbf)- which is the same as opening the file. It then sets the format to epicc (epicc.fmt), which is a customized screen for the contractor records. The command "edit" tells the computer to allow the user to edit the records, by positioning him or her on the first record in the file, from which the user can scroll through the whole dBase file. The rest of Case 1 tells the computer to close epicc, clear the screen, and return to the menu.

Case 2 does the same thing as Case 1, except it tells the computer to append the epicc file. It places the user on a blank record which is directly after the last record in the file. The user can then fill in that record with new information, and the record is automatically added to the file. The user can add as many files as he or she wishes before returning to the menu, and he or she can also scroll back through all of the old records.

Case 3 tells the computer to open the contractor file, then change the output device from screen to printer. It then tells the computer to do the following commands until it reaches the end of the file ("do while .not. eof()"). Lines 46-80 tell the printer exactly what to print on the page, and 81, "skip", tells the computer to go to the next record when it reaches the end of the record it is printing. Then the device is set back to the screen, and the file is closed, the screen cleared, and the computer returns to the menu.

Case 4 sends the computer to a subprogram, "name", which tells it how to search for a specific company name. A description of name.prg can be seen following the hard copy of that program. Case 5 also sends the computer to another program, dbepic.prg, which will bring the main menu up to the screen. The user can then end the editing session, or go on to another subprogram, such as epicu.prg or epicg.prg. Again the commands "end do" end and close the do loops and the program.

EPICU.PRG

```
set talk off                                                    1
set bell off
set escape off
set status off
choice=" "                                                      5

do while .t.

  choice = " "
  @ 3,3 to 15,46 double                                        10
  @ 5,13 say "University File Manager"
  @ 7,15 say "1.Edit Files"
  @ 8,15 say "2.Append Files"
  @ 9,15 say "3.Print Files"
```

```
@ 10,15 say "4.Search by University Name"
@ 11,15 say "5.Return to Main Menu"
@ 13,15 say "ENTER OPTION:"
@ 13,29 get choice
read

do case

case choice = "1"
   use epicu
   set format to epicu
   edit
   close all
   clear all
   clear
   loop

case choice = "2"
   use epicu
   set format to epicu
   append
   close all
   clear all
   clear
   loop

case choice = "3"
   use epicu
   set device to print
   do while .not. eof()

     @ 3,5 SAY "UNIVERSITY NAME:"
     @ 3,21 SAY UNAME
     @ 5,5 SAY "ADDRESS:"
     @ 5,30 SAY "UNIVERSITY POC:"
     @ 5,55 SAY "GOVERNMENT POC:"
     @ 6,5 SAY ADDRESS1
     @ 6,30 SAY UPOC1
     @ 6,55 SAY GPOC1
     @ 7,5 SAY ADDRESS2
     @ 7,30 SAY UPOC2
     @ 7,55 SAY GPOC2
     @ 8,5 SAY ADDRESS3
     @ 8,30 SAY UPOC3
     @ 8,55 SAY GPOC3
     @ 9,5 SAY ADDRESS4
     @ 9,30 SAY UPOC4
     @ 9,55 SAY GPOC4
     @ 11,5 SAY "CONTRACT?"
     @ 11,14 SAY CONTRCTNO
     @ 11,35 SAY "RELEASE FORM:"
     @ 11,49 SAY REL1
     @ 12,5 SAY "VERSION:"
     @ 12,13 SAY VERSION
```

```
            @ 14,5 SAY "CORRESPONDENCE:"
            @ 15,5 SAY COR1                                    70
            @ 16,5 SAY COR2
            @ 17,5 SAY COR3
            @ 18,5 SAY COR4
            @ 19,5 SAY COR5
            @ 20,5 SAY COR6                                    75
            @ 21,5 SAY COR7
            @ 22,5 SAY COR8
            @ 23,5  SAY "NOTES:"
            @ 24,5 SAY NOT1
            skip                                               80
         enddo
         set device to screen
         close all
         clear all
         clear                                                 85
         loop

      case choice = "4"
         clear all
         close all                                             90
         clear
         do name3

      case choice = "5"
         clear all                                             95
         close all
         clear
         do dbepic

      enddo                                                   100

   enddo
```

EPICU.PRG does basically the same thing as epicc.prg. The only difference is the file that it gives access to: the university file. The menu is slightly different, as are the commands which tell the computer what to print on paper in Case 3. This program also has its own subprogram, name3.prg, to give it its searching ability. Other than those minor differences, epicu.prg is the same as epicc.prg.

EPICG.PRG

```
set talk off
set bell off
set escape off
set status off
choice = " "

do while .t.

  choice = " "
  @ 3,3 to 15,46 double
  @ 5,13 say "Government File Manager"
  @ 7,15 say "1.Edit Files"
  @ 8,15 say "2.Append Files"
  @ 9,15 say "3.Print Files"
  @ 10,15 say "4.Search by Name"
  @ 11,15 say "5.Return to Main Menu"
  @ 13,15 say "ENTER OPTION:"
  @ 13,29 get choice
  read

  do case

  case choice = "1"
     use epicg
     set format to epicg
     edit
     close all
     clear all
     clear
     loop

  case choice = "2"
     use epicg
     set format to epicg
     append
     close all
     clear all
     clear
     loop

  case choice = "3"
     use epicg
     set device to print
     do while .not. eof()

       @ 3,5 SAY "INSTALLATION:"
       @ 3,18 say INSTAL
       @ 4,5 SAY "BRANCH:"
       @ 4,12 say branch
       @ 5,5 SAY "ADDRESS:"
```

```
                @ 5,36 SAY "POC:"
                @ 6,5 say ADDRESS1
                @ 6,36 SAY POC1
                @ 7,5 say ADDRESS2
                @ 7,36 say POC2
                @ 8,5 say ADDRESS3
                @ 8,36 say POC3
                @ 9,5 say ADDRESS4
                @ 9,36 say POC4
                @ 11,5 SAY "RELEASE FORM:"
                @ 11,18 say REL1
                @ 12,5 SAY "VERSION:"
                @ 12,13 say VERSION
                @ 14,5 SAY "CORRESPONDENCE:"
                @ 15,5 say COR1
                @ 16,5 say COR2
                @ 17,5 say COR3
                @ 18,5 say COR4
                @ 19,5 say COR5
                @ 20,5 say COR6
                @ 21,5 say COR7
                @ 22,5 say COR8
                @ 23,5 SAY "NOTES:"
                @ 24,5 say NOT1
                skip
             enddo
             set device to screen
             close all
             clear all
             clear
             loop

          case choice = "4"
             clear all
             close all
             clear
             do name2

          case choice = "5"
             clear all
             close all
             clear
             do dbepic

          enddo

       enddo
```

EPICG.PRG is also basically the same as epicc.prg, except for a few small details, such as screen format, printout format, and its subprogram, name2.prg.

NAME.PRG

```
set talk on                                              1
set bell off
set escape off
set status off
m_coname= space(50)                                      5

do while .t.

  m_coname = space(50)
  @ 3,3 to 7,75 double                                  10
  @ 5,10 say "Enter Name:"
  @ 5,21 get m_coname
  read

use epicc                                               15
set format to epicc
edit for coname = m_coname
close all
clear all
clear                                                   20
do epicc
```

Lines 1-4 of name.prg set the program environment, and line 5 of the program initiate a memory variable of up to fifty characters. Line 7 opens a do loop, which creates a screen which requests that the user enter a company name. The computer assigns that name to a temporary memory variable, in this case, m_coname. It then opens epicc, sets the format to epicc, and searches for the file whose coname (company name) equals the memory variable m_coname. Once it finds that record, it brings it up on screen, where the user can edit it. If the computer cannot make a match, it returns to the menu of epicc. After the user is done editing, and hits the Esc key, the computer saves all edits and returns to the epicc menu.

NAME2.PRG

```
set talk on
set bell off
set escape off
set status off
m_instal= space(50)

do while .t.

  m_instal = space(50)
  @ 3,3 to 7,75 double
  @ 5,10 say "Enter Name:"
  @ 5,21 get m_instal
  read

use epicg
set format to epicg
edit for instal = m_instal
close all
clear all
clear
do epicg
```

The name2.prg is the same as name.prg, except it is tailored to the needs of the government file.

NAME3.PRG

```
set talk on
set bell off
set escape off
set status off
m_uname= space(50)

do while .t.

  m_uname = space(50)
  @ 3,3 to 7,75 double
  @ 5,10 say "Enter Name:"
  @ 5,21 get m_uname
  read

use epicu
set format to epicu
edit for uname = m_uname
```

```
close all
clear all
clear                                                    20
do epicu
```

NAME3.PRG is the same as name.prg and name2.prg, except it is the subprogram of epicu.prg.

This concludes the documentation of the EPIC Distribution Database. The follwing pages contain a list of the names of contractors, universities, and government agencies which must be used to successfully use the search option of this program.

VI. Conclusion

In conclusion, this summer has been a wonderful learning experience for me. I learned how to program computers, work in an office environment, and prepare for life in college and the workplace. My project has been completed, and the transfer of the management of EPIC distribution can be handled easily and quickly, so that Mr Nixon will be able to go to school without any worry about it. The documentation is also complete, so that the new EPIC distribution manager will have no problems using my database program.

VII. Other Lessons of the Summer

Not only did I learn how to program in dBase this summer, I also learned Microsoft QuickBASIC and C. I learned how to make nine-track tapes for the distribution of the EPIC hydrocodes using VAX systems, and how to operate those systems to run data decks using EPIC hydrocode calculations. I also learned about

what all the other apprentices were doing during the summer, through our discussions at lunch and all the free time that we spent together this summer. Not only did my summer job give me work experience, it gave me many friends who are almost like family to me now. If I had the chance to do it all over again, I'd ask "When do I start?"

VIII. Bibliography

Ashton-Tate dBase III Plus. (Documentation). Ashton-Tate, St. Joseph, Michigan, 1987.

Johnson, G R and R A Stryk. User Instructions for the 1989 Version for the EPIC-2 Code. Honeywell, Inc, Brookland Park, MN, June 1990.

Micropro Wordstar Professional. Micropro International Corporation, Benton Harbor, Michigan, 1987.

Microsoft BASIC, Professional Development System, BASIC Language Reference. Microsoft Corporation, 1989.

Microsoft BASIC, Professional Development System, Programmer's Guide. Microsoft Corporation, 1989.

SYNTHESIS AND CHARACTERIZATION

OF

3-PICRYLAMINO-1,2,4-TRIAZOLE

Kathryn Deibler
Mr Stephen A. Aubert, Mentor
August 15, 1990

INTRODUCTION
Section I

3-picrylamino-1,2,4-triazole (PATO) is a heterocycle under investigation as a possible high energy insensitive explosive. Preparation of PATO was conducted in 10 g and 100 g scales. The two syntheses were verified using instrumental analyses. Physical, structural, sensitivity, and explosive characteristics of PATO were determined.

CONTENTS

I. Introduction

II. Acknowledgments

III. Background

IV. Procedures

V. Results

VI. Conclusion

VII. Miscellaneous

VIII. References

ACKNOWLEDGMENTS
Section II

I would like thank Mr Stephen A. Aubert, Dr Robert L. McKenney, and Mr Thomas G. Floyd for designing a project for me which involved interaction with several different chemistry instruments. I appreciate Dr Michael A. Patrick's instructions on the mass spectrometer, particle size analyzer, scanning electron microscope, x-ray spectrometer, and Perkin DSC; Mr Floyd and Mr Stevens' help with the Dupont 2100 Thermal Analyst; Dr McKenney's instructions on the infrared spectrometer and general assistance when my mentor was not available; Dr Paul R. Bulduc's assistance on the computer and lessons on chemistry theory; and Lt Joseph F. Ford's running elemental analysis. I appreciate Mrs Lois Walsh's assistance in preparing my presentation. I would like to thank Mr Don D. Harrison and Dr Norm Klasutous for organizing the HASP. A special thanks to Mr Stephen A. Aubert for teaching me the chemistry background of my project and for his time and patience throughout the summer.

BACKGROUND
Section III

Accidental initiation of 1.1 hazard class munitions presents a great threat to United States Air Force facilities. The High Explosive Research and Development facility at Eglin Air Force Base is developing high energy insensitive explosives to alleviate this threat. Insensitive high explosives offer advantages in operational readiness and sustainability through reduced quantity distance storage and handling restrictions (reference 1). Munitions filled with such explosives may be stored closer to airfields where they are readily available for use. Transportation of these munitions is also less hazardous. Insensitive munitions have a low probability of reaction when subjected to thermal, impact, friction, or electrostatic stimuli enhancing their surviablity. However, the explosive will detonate with great power upon initiation of its designed explosive train. Most importantly they have low shock sensitivity and hence are non-mass detonating.

Explosives are useful militarily for applying disruptive energy to a target by converting chemical energy into kinetic energy. Chemical energy is applied by imparting kinetic energy into high velocity case fragments and expanding product gases. The combination of high energy output and insensitivity to shock initiation are highly desirable but difficult to obtain simultaneously with the present energetic materials available and conventional explosive formulation techniques.

Heterocycles are organic ring compounds with atoms other then carbon making up the skeletal ring (reference 3). Azoles are five membered heterocycles with one or more nitrogens. Heterocylcic materials are ideal for insensitive high explosives. Replacement of

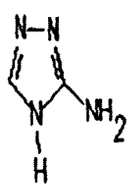
carbon with nitrogen provides the advantages of a higher density, favorable oxygen balance, and a positive heat of formation. High density improves a materials energy per unit volume, and hence its performance. Fuel, carbon and hydrogen, in an explosive reacts by combining with oxygen, producing carbon dioxide and water. A favorable oxygen balance, where all carbon and hydrogen is consumed with oxygen, results in more efficient reactant to product conversion and hence greater energy release. In the heterocycle, 3-picrylamino-1,2,4-triazole (**PATO**), nitrogens replace the carbons that would not have reacted due to an unfavorable carbon oxygen ratio. In addition aromatic nitro linkages decrease the compounds' sensitivity. High energy from the favorable oxygen balance, high density, favorable heat of formation, and insensitivity from the aromatic nitro linkage make heterocycles promising candidates for insensitive explosive applications.

3-Picrylamino-1,2,4-Triazole (**PATO**) was first produced at the Los Alamos National Laboratory (LANL), and was characterized as a high density insensitive material (reference 2). Its energy was insufficient to meet nuclear applications, hence, only preliminary testing was completed. The triazole in **PATO** gives it the advantages of a heterocycle. **PATO** is synthesized by amination of picryl chloride (2,4,6-Trinitrochlorobenzene) with 3-amino-1,2,4-triazole (**ATA**) in dimethylformamide (DMF) at 100 degrees Celsius for a period of five hours (reference 2). The aryl halide (Ar:X), picryl chloride, and the amine (Z), **ATA**, undergo a two step bimolecular displacement nucleophilic aromatic substitution (reference 3).

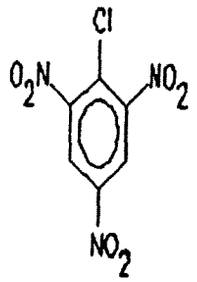
$$Ar{:}X + :Z \longrightarrow Ar\overset{:X^-}{:Z}$$

$$Ar\overset{:X^-}{:Z} \longrightarrow Ar{:}Z + :X^-$$

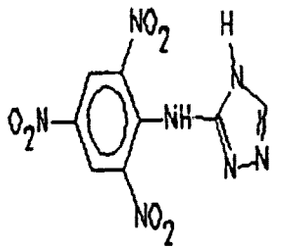
An aryl halide contains an aromatic ring (Ar) with an attached halogen (X), an element in group VIIa on the Periodic Table. An aromatic compound is an organic ring with alternating double bonds. Orbital overlap of the alternating double bonds results in a resonance structure forming a delocalized pi cloud of electrons. Benzene carbon-carbon bond lengths are all equivalent and half the distance between single and double bond length. The heats of hydrogenation and combustion of benzene are lower than expected. These physical properties are a result of the delocalized resonance structure. Picryl chloride (2,4,6-trinitrochlorobenzene) is the aryl halide used in the synthesis of **PATO**. Picryl chloride's nitro groups (NO_2) located in the ortho and para positions of the benzene ring (2 and 5) are strongly electron withdrawing (reference 3).



ATA



Picryl chloride



PATO

The first step of the two step mechanism is the attack of the nucleophile, **ATA**, on the aryl halide bond. The nucleophile has a lone pair of electrons located at the primary nitrogen. The first step forms a carbanion which is a negatively charged aromatic ring, halogen, and amine. This step is the slow, rate limiting and rate determining step.

STEP 1

$$C_6H_5Cl + :ATA \rightarrow {}^{-}C_6H_5 < \begin{matrix} Cl \\ ATA \end{matrix} + H^+$$

Nitros attached to the aromatic ring increase the rate of formation of the carbanion by dispersion of its negative charge. Nitros, nitrides, acid, and aldehyde attachments are electron withdrawing and delocalize the negative charge further stabilizing the carbanion. The carbanion is a full negative ion formed by the amine attaching at the same bond as the halide. During the second step the halogen, chlorine, dissociates from the aromatic ring with a hydrogen forming hydrochloric acid and the aryl amine (reference 3).

STEP 2

$$_C_6H_5 < \begin{matrix} Cl \\ ATA \end{matrix} + H^+ \rightarrow C_6H_5ATA + HCl$$

The preparation of aniline from chlorobenzene is well known. $C_6H_5Cl + NH_3$ under 15 PSI at 1500^oC yields $C_6H_5NH_2 + HCl$. The preparation of nitroaniline from nitrochlorobenzene is known to occur much more readily. $C_6H_4NO_2Cl + NH_3$ in DMF at 100^oC yields $C_6H_4NH_2NO_2 + HCl$. **PATO** with two more nitro groups can be expected to react more readily.

PROCEDURES
Section IV

PATO was synthesized by reacting 3-amino-1,2,4-triazole, **ATA** (16.77g), and picryl chloride (23.34 g) in 170 mL of N,N-dimethylformamide (DMF) in a 1500 mL round bottom flask. The solution was heated at 100°C using a heating mantle and variac control for five hours. 1000 mL of water near 0°C was poured into the solution after the completion of the reaction to induce precipitation of the **PATO**. The precipitated solid was collected by vacuum filtration. The precipitate was dried in a Napco model 5831 vacuum oven for 24 hours. The reaction equation and stoichiometry are as follows:

$C_2H_6N_4$ + $C_6H_2N_3O_6Cl$ --> $C_8H_5N_7O_6$ + HCl

.176 mol + .068 mol --> .068 mol + .068 mol

16.77 g + 23.33 g --> 20.00 g

The large scale synthesis of **PATO** was accomplished using a Mettler RC1 reaction calorimeter. In this scale reaction 72.93 g **ATA** and 83.89 g of picryl chloride in 800 mL of DMF was heated at 100°C for five hours. Precipitation of **PATO** was induced by adding 2000 mL of cold water to the solution. The solution was vacuum filtered and the product was dried in a vacuum oven at 60°C. The reaction equation and stoichiometry are as follows.

$C_2H_6N_4$ + $C_6H_2N_3O_6Cl$ --> $C_8H_5N_7O_6$ + HCl

.3388 mol + .847 mol --> .3388 mol + .3388 mol

83.87 g + 72.92 g --> 100 g

(reference 2)

Recrystallization was accomplished by dissolving 10 g of **PATO** in 100 mL of cyclohexanone in a 400 mL beaker. The temperature was gradually raised to 100°C.

Elemental analysis, energy dispersive x-ray microanalysis, and nuclear magnetic resonance (NMR), infrared, and mass spectrometry were used for identification and quantitative evaluation of the product material. The physical characterization of **PATO** was accomplished using a scanning electron microscope and a particle size analyzer. Thermal analysis was conducted on a differential scanning calorimeter and a Henkin time-to-explosion tester. Impact sensitivity was determined using a drop hammer.

The mass spectrum of **PATO** was obtained with a Finnagen MAT 5100 GC/MS system at 340°C using a direct inlet port. A mass spectrum can give an exact molecular weight. The number of molecular formula possibilities are narrowed and the presence of certain structural units can be detected and identified. A mass spectrum shows the signal obtained from the fragment ions' mass to charge ratios (reference 4).

Proton NMR spectra were obtained using a Bruker AC-300, 300 MHz Fourier Transform Superconducting NMR spectrometer. A 3 mg sample of **PATO** dissolved in 1 mL of deutero DMF was placed in a 5 mm glass NMR tube. Each structure's protons experience a characteristic applied field strength which produce a set effective field strength at which absorption takes place. The resonance frequency (ν) is determined by the fundamental NMR equation $\nu = \delta / 2\pi H^{o}$, where δ is the gyromagnetic ratio and H^{o} is the main field strength. Nuclear magnetic resonance spectroscopy measures the applied field strength plotted against the absorption signal. In a NMR a superconducting magnet produces a homogeneous magnetic field of approximately 7.4 Tesla between its poles. The sample is spun about its main field (Z) axis by a stream of air to average out any existing homogeneities in the xy plane. The number of

signals on the spectrum tells how many "kinds" of structure there are in the molecule. The position of the signals tells about the electronic environment of each proton. The intensity of the signals shows the number of protons of each kind that exist. The splitting of a signal into several peaks shows the environment of a proton with respect to other, nearby protons (reference 3).

Elemental analysis for carbon, hydrogen, and nitrogen was conducted on a Carlo Erbe, Model No. 1106, CHNO analyzer. The sample was crimped in a tin container and dropped into a quartz tube at 1030°C. The elemental analysis quantitatively analyzes the weight percent of carbon, hydrogen, and nitrogen in a sample. The theoretical percents for carbon, hydrogen, and nitrogen in **PATO** are 32.55, 1.71, and 33.22, respectively (reference 2).

An infrared spectrum was obtained using a Mattson Cygnus 25 Fourier Transform-Infrared (FT-IR) spectrometer. It is characteristic of organic groups to absorb infrared radiation of certain frequencies. This radiation absorption is converted by the molecule into energy of molecular rotation. The absorption is quantized, creating the infrared spectrum (reference 4).

Every element in a sample will emit a unique and characteristic pattern of x-rays. The x-rays emitted by **PATO** molecules were quantized in a X-ray analyzer attached to a DS-130S Scanning Electron Microscope. The sample was coated with gold to make the sample conductive (reference 5). Concentration of the elements could not be determined as a consequence of the gold coating.

Particle size was determined by a Brinkmann Model 2010 Particle Size Analyzer. Two samples were suspended in glycerin with a 98.5%

concentration in a magnetic cell. The particle size was measured using the scanning laser technique.

PATO, ATA, and picryl chloride (1 mg samples) were thermally analyzed on a Perkin-Elmer System-4 Differential Scanning Calorimeter and a DuPont 2100 Thermal Analyzer. Heat was applied at a rate of 10.00 degrees per minute. The scan produces a trace of the endothermic and exothermic events which take place as the temperature of the sample rises in an inert N_2 environment (reference 6). The samples were crimped in either a vented or air tight aluminum sample pans.

The LASL method for determining critical temperature based on time-to-explosion tests was used (reference 7). A 40 mg explosive sample was pressed in a DuPont E-83 aluminum blast-cap shell and sealed by a 400 lb force applied to a hollow skirted plug. The highest temperature at which four samples do not detonate after 2000 seconds (no-go) is concluded to be the critical temperature. Tests were conducted at 340°, 330°, 320°, 310°, 300°, 295°, 290°, 287°, 285°, 284°, 283°, 282°, 281°, and 280°C. The apparent energy of activation and frequency factor (^Z) were determined by plotting the log of time against the inverse of the temperature. The critical temperature (Tm) can be calculated for a given size in terms of the kinetic and physical parameters (reference 8).

Frank Kaminetskii Equation, EQ 1

$$Tm = \frac{E}{R \ln \frac{A^2 \rho Q Z E}{Tm^2 \lambda \delta}}$$

where:

R = gas constant, 1.987 cal/mol
A = half thickness
ρ = density
Q = heat of decomposition
Z = pre-exponential factor
E = activation energy
λ = thermal conductivity
δ = shape factor (2.0 for infinite cylinder)

Impact data were obtained with a Bureau of Mines, Model No.2, Drop Hammer. Twenty 35 mg samples were tested with a 5 kg weight dropped from 200.5 cm. Each powdered sample was impacted on Norton coated abrasive paper of 180 grit (FSN 5250-271-7930). The impact sensitivity $H_{50\%}$ was obtained using a Bruceton up down procedure.

RESULTS
Section V

When ATA, picryl chloride, and DMF were added in the flask, the liquid gradually turned dark maroon. Small yellow crystals formed and fell to the bottom as the cold water was added. The first synthesis of PATO had a wet weight of 44.236 g. After drying the filtrate in a vacuum oven the weight was 12.04 g, a 62.25 percent yield. The scale-up reaction produced a 226.05 g wet weight, with a 86.73 g dry weight, a 86.73 percent yield. Recrystallization was unsuccessfully attempted using cyclohexanone. Only 3 g of PATO were soluble in 100 mL of cyclohexanone and would not precipitate from the solution. Since recrystallization of PATO was unsuccessful, scans and spectra were derived from the unrecrystallized material.

No molecular ion was observed on the mass spectrum (figure 1). The peak of greatest mass was exhibited at 203 atomic mass units (amu), which corresponds to the loss of two nitro groups ($-NO_2$). The next peak was located at 157.2 amu indicating the loss of the third nitro group. Peaks at 145.2, 130.2, and 116.1 amu show the fragmentation of the triazole ring with the additional loss of a carbon, a nitrogen and hydrogen, and a nitrogen, in respective order. The peak at 103.1 amu suggests a rearrangement of the bonds with the loss of another carbon. This fragmentation pattern is consistent with that expected of PATO.

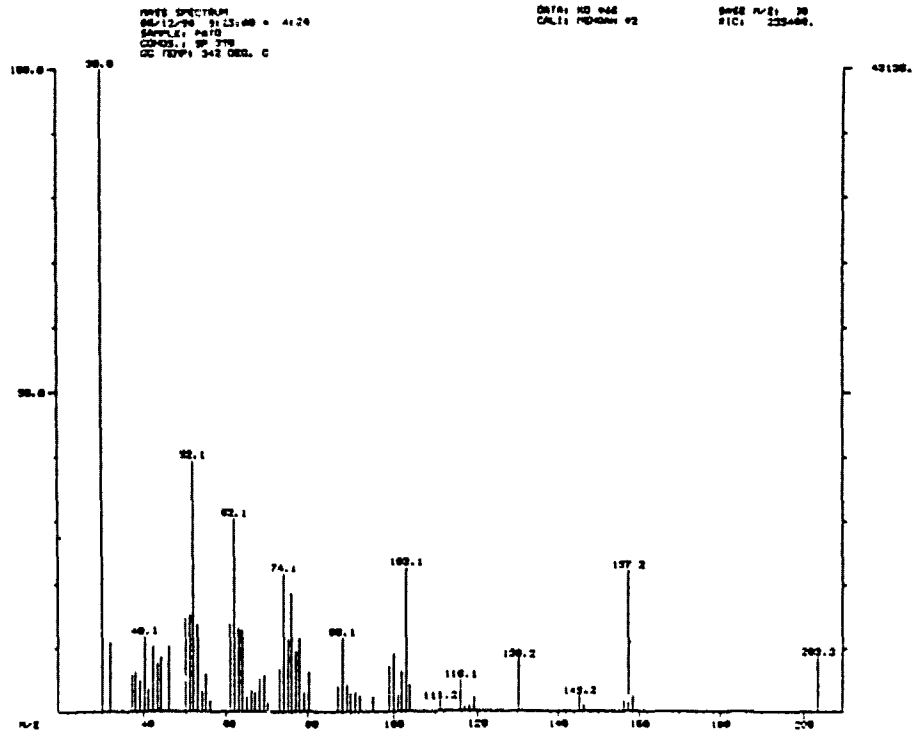


FIGURE 1 Mass Spectrum

The Proton NMR Spectrum (figure 2) contained one single peak at a field shift of 9.13 ppm. This peak corresponds to the absorption of the aromatic hydrogen and was shifted down in field in comparison with picryl chloride indicating successful conversion to PATO.

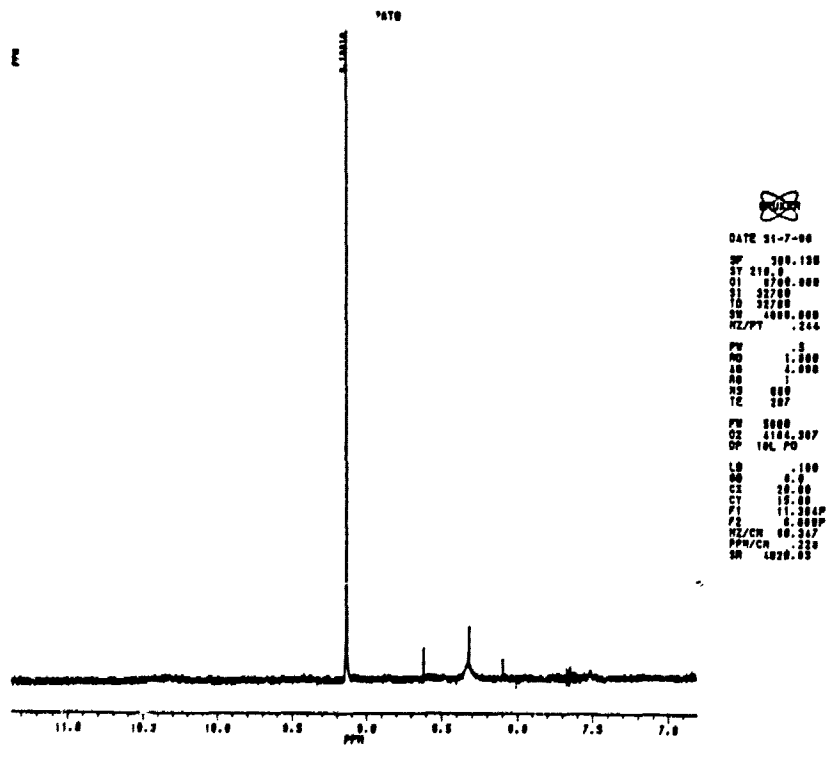


FIGURE 2 NMR Spectrum

Elemental analysis mean percent composition for carbon, hydrogen, and nitrogen was 31.93 +/- .08, 1.825 +/- .005, and 32.97 +/-.06, respectively. The measured results were in good agreement with the theoretical values, indicating the successful synthesis of **PATO.** The percentages from each analysis were as follows:

	Theoretical	Run 1	Run 2	Run 3	Mean +/-	Error
C	32.55	31.7884	31.9645	32.0451	31.93 .08	.62
H	1.71	1.8162	1.8236	1.8340	1.825 .005	.115
N	33.22	32.8684	33.0778	32.9782	32.97 .06	.25

The peaks indicating infrared radiation absorption on the infrared spectrum (figure 3) correspond with the infrared radiation produced by the bond structures in **PATO.** The presence of the following organic groups were verified by the infrared spectrum:

PEAK LOCATION (wavenumbers)	BOND INDICATED
3300-3500	N-H Amine
1173, 1204, 1237, 1313, 1348	C-N Amine
1619, 1595	-NH Amine
1529, 1348, 1383, 542, 520	-NO 2 Nitro
861, 736, 717	Benzene with symmetrical atch
924, 980	Alkene

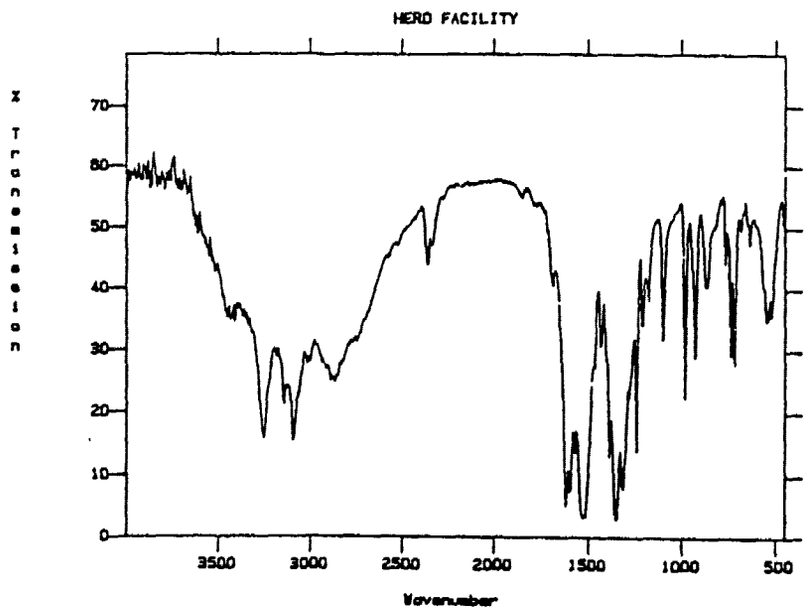


FIGURE 3 Infrared Spectrum

The X-ray spectrum (figure 4) showed the presence of carbon, nitrogen, oxygen, and a trace amount of chlorine. The mean particle size obtained from the particle size analyzer was 1.26 microns. The number distribution is displayed in the below table.

PARTICLE SIZE

Under (%)	Size (microns)	Over (%)
55	0.97	45
85	1.50	15
90	1.89	10
95	2.95	5
97	3.92	3
99	5.55	1
100	114.50	0

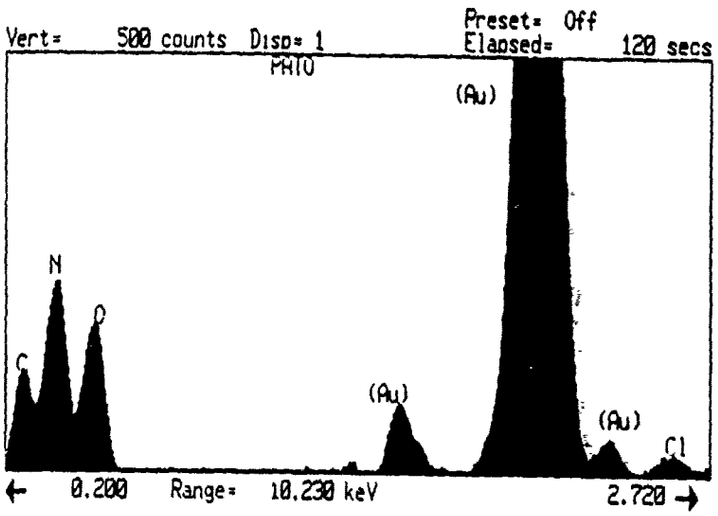


FIGURE 4 X-ray Spectrum

The onset of the exotherm on the differential scanning calorimeter (DSC, figure 5) varied from 310°C to 320°C. The decomposition temperatures, onset temperatures and area under the curve obtained from the DSC were as follows:

DSC RESULTS

Material	Conditions	Exotherm	Onset	Area
PATO	vented Al pan,	322.97°C	319.69°C	178.3 cal/g
PATO	unvented Al pan,	319.05	315.51	188.42
PATO	N2 purge Al pan,	314.15	310.09	105.0
		Endotherm		
ATA	unvented Al pan	155.38°C	152.65	
Picrylchloride	"	78.84	75.69	

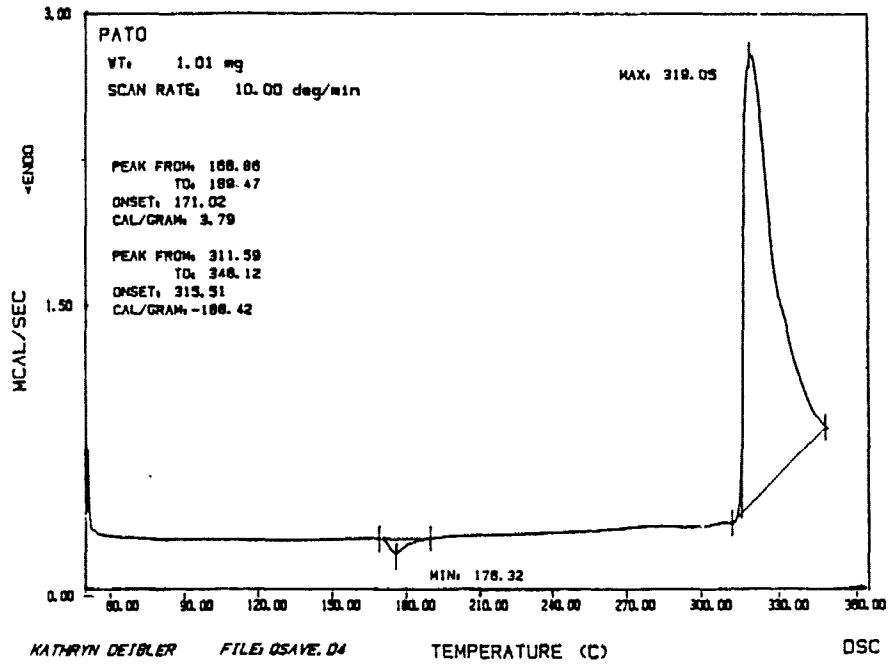


FIGURE 5 DSC

The critical temperature (Tm) for **PATO** was measured at 282 +/- 1°C using the Henkin time to explosion test (figure 8). The energy of activation and frequency factor calculated to be 23 kcal and 3.4×10^{-8}, respectively (figure 7). Four no-go tests were recorded at 282 degrees Celsius. No-go's were obtained at 280, 281, and 282 degrees Celsius. The times associated with each temperature are listed below.

HENKIN TIME TO EXPLOSION RESULTS

Temperature (°C)	Time (s)	Temperature (°C)	Time (s)
340	6.8	287	31.1
330	8.1	285	38.3
320	8.2	284	35.35
310	9.5	283	55.8
300	13.9	282(4 runs)	2000 no-go
295	15.7	281	2000 no-go
290	24.4	280	2000 no-go

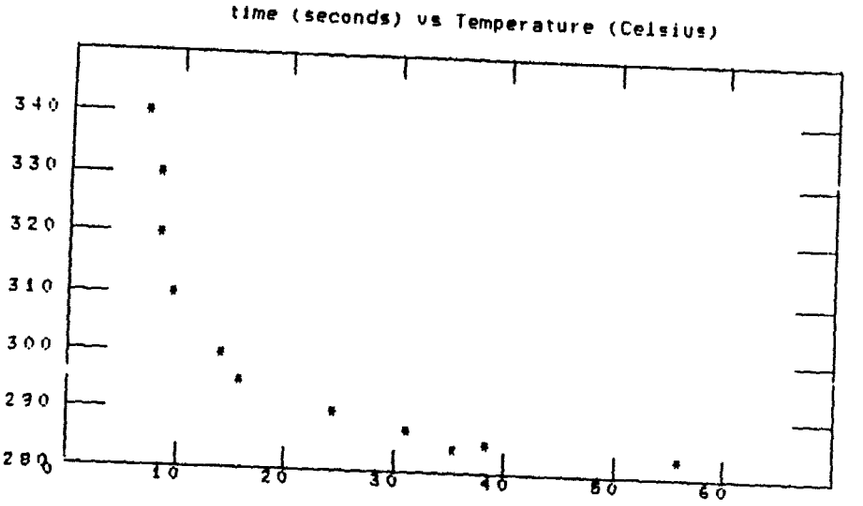


FIGURE 6, Henkin Time to Explosion Results

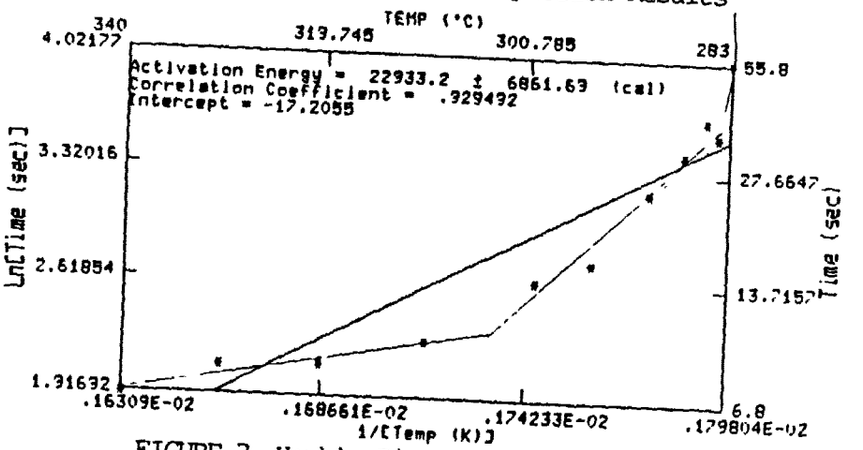


FIGURE 7, Henkin Time to Explosion Calculations

Impact sensitivity tests resulted in no reaction of **PATO** in 20 drops at a height of 200.5 cm with a weight of 5 kilogram. The impact sensitivity height of 50 percent probability of reaction ($H_{50\%}$) is therefore greater than 200.5 cm. This is similar to **TATB**, the standard for insensitivity.

CONCLUSIONS
Section VI

The results from the mass spectrum, NMR scans, elemental analysis, infrared spectrum, scanning electron micrograph, and x-ray spectrum verify the compound synthesized is **PATO.** **PATO** is thermally stable with a 318.72 degrees Celsius decomposition exotherm and a critical temperature of 282°C. Its drop weight impact sensitivity with a 5 kg weight is greater than 200.5 cm, extremely insensitive. It does have an unfavorable small particle size which can probably be overcome through recrystallization. **PATO** is thermally stable, insensitive to impact, and high in performance making it a promising candidate as an insensitive high explosive.

MISCELLANEOUS
Section VII

This summer through the HSAP I learned the importance of patience and precision in conducting scientific experimentation. I found that repetition of similar tests are necessary for accuracy. I gained experience in applying the scientific method and realized the importance of keeping accurate and detailed records. I had the opportunity to use advanced chemistry instruments that most students studying chemistry will not get to use until after graduation from college.

REFERENCES
Section VIII

1. Aubert, Stephen A. Interim Qualification of AFX-1100, AFATL-TR-89-48, Air Force Armament Laboratory, Eglin AFB, Florida, September 1989.

2. Coburn, Michael D. 3-Picrylamino-1,2,4-Triazole and its Preparation. U.S. Patent 3 483 211, 1969.

3. Morrison, Robert T. and Robert N. Boyd. Organic Chemistry, Third Edition. Boston: Allyn and Bacon, Inc., 1973.

4. Silverstein, Robert M., G. Clayton Bassler and Terence C. Morrill. Spectrometric Identification of Organic Compounds, Forth Edition. New York: John Wiley and Son, 1981.

5. Energy-Dispersive X-ray Microanalysis, An Introduction. San Carlos, California: Kevex Corporation, 1988.

6. Perkin-Elmer. Instructions to Perkin-Elmer Model 3600 Data Station PETOS OS/12. Norwalk, Connecticut: Perkin-Elmer, 1983.

7. Gibbs, Terry R. and Alphonse Poplatoetal. "LASL Explosives Property Data." Berkeley, California: University of California Press, 1980.

8. Dobratz, B. M. "LLNL Explosives Handbook Properties of Chemical Explosives and Explosive Simulants." Livermore, California: University of California, March 16, 1981.

NEURAL TARGET IDENTIFICATION

Christopher Ellis

10 August 1990

OUTLINE

1. Introduction
   - 1.1 Section description
   - 1.2 Section functions
   - 1.3 My research within this function

2. Neural Network Introduction
   - 2.1 Description
   - 2.2 Operation
   - 2.3 Neural Nets and target ID

3. Description of Research
   - 3.1 Purpose
   - 3.2 Procedure
   - 3.3 Results

4. Conclusions and Acknowledgments
   - 4.1 Neural Net advantages
   - 4.2 Acknowledgments

INTRODUCTION

This paper contains the final research results of Christopher Scott Ellis in the High School Apprenticeship Program (HSAP) for the eleven-week term of the summer of 1990. All work was done at the Eglin Air Force Base Armament Laboratory, in the Advanced Guidance Instrumentation Branch, Radar and Image Simulation Section (AGI/RISS). This work was performed under the mentorship of Mr. Mike Wallace, the Section Chief of RISS.

The RISS operates two computer labs located on the fourth floor, the Radar Signal Processing Lab (RSPL), and the Image Processing Lab (IPL). Both labs boast VAX mainframes, disk farms, and numerous hi-res RGB monitors. The vast majority of my number-crunching needs were satisfied in these labs. My terminal was a VAX Workstation, linked to a MicroVax, in my mentor's office. I also utilized two thermal wax printers to make hardcopies and view-graphs of images on the RGB monitors. In addition, primarily for generating my report and seminar, I utilized a Z-248 with the HP LaserJet II and QMS ColorScript 100 printers.

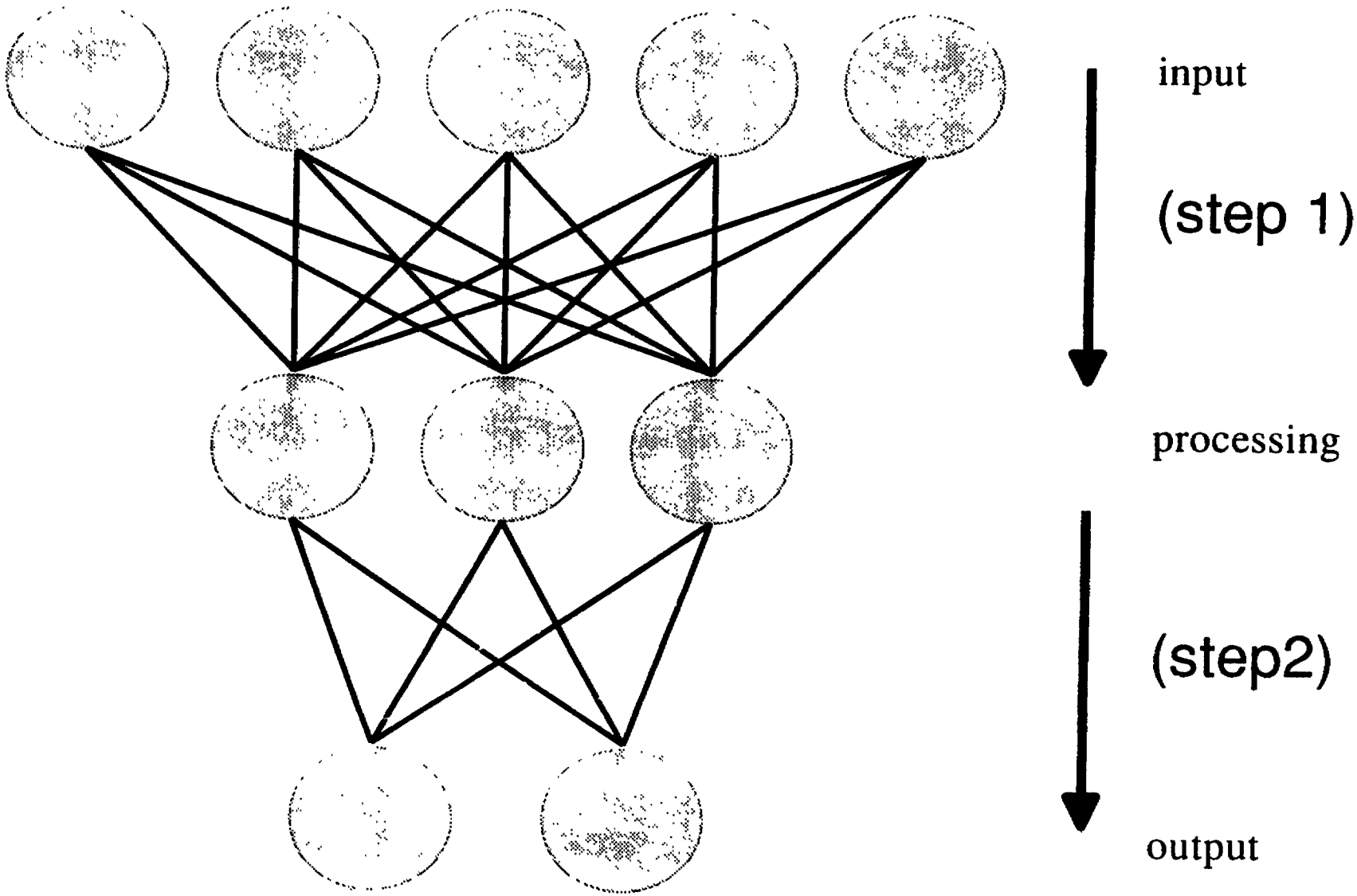
The work done in the RSPL and the IPL centers on image processing. Specifically, using image processing techniques to take images from missile seekers and extract targets from them. Traditionally, this was done by utilizing image algebra, as well as various techniques such as thresholding, transforms, and edge detection. My task this summer was to utilize the technology of neural networks to perform extractions, as opposed to the said techniques.

Neural Network (NN) Introduction

A NN is simply a cluster of interconnected neurons which are programmed to solve a particular problem. Each neuron is a very simple processor. It receives impulses from a number of other neurons, adds them up, passes the value through some function, and then passes that value to a number of other neurons. The network architecture I used was the Back Propagation (BP) arrangement, as shown in the accompanying diagram. Each neuron in the input layer is stimulated to a value, representing to some parameter of the problem. The impulses travel trough the NN to the middle then the output layer, where the outputs of the output neurons represent some part of the NN's solution.

A NN is functionally the same as our own built in NN, the brain; it associates, and identifies, patterns. A NN is programmed to perform a task by teaching it a set of inputs and corresponding outputs. For instance, a network might associate the 26 8x8 bitmap images of the ASCII letters with the actual ASCII code (an OCR application). Similarly, a tank can be trained on an image of a tank, or an armored personnel carrier (APC), and then recognize another in a missile seeker's scene.

Example Network



Description of Research

With this enormous power of networks in mind, I set forth with the following research goals:

1) Produce the code to train a BP network
2) Produce the code to scan a 512x512 image with a network trained on 50x50 images
3) Obtain intensity-altitude plot of the NN's outputs

The first two steps were the most time consuming, as well as the most productive, resulting in 12 megs of code, data, and image files.

I now began my experiments, using the accompanying digitized picture as a test. I trained the network to recognize the APC located in the upper left area. I wanted to see two things: one, if the network would also identify the two alternate targets, thus showing adaptive behavior, and two, if the NN rejected the noisy, cluttered background that is deceptively like a target.

The accompanying graph shows the results of the NN's scan of the image. Areas of high intensity are where the NN recognized the APC which it trained upon. Conversely, areas of low intensity were not recognized as a probable tank. The 3D plot is an altitude plot of intensity. Peaks represent areas of intensity.

As can be seen, the NN passed both of my tests. First, it gave high responses for the other two targets, which look nothing like the APC. Next, it rejected areas of vegetation which are the same size and intensity as the targets.





Conclusions and Acknowledgments

The NN used in the experiment demonstrated several advantages of NNs to traditional techniques (TT). TTs simply identify objects against the background. Although a tank certainly fits this description, so does a large tumbleweed. In addition, if the target is covered by a net, or is otherwise camouflaged (in the visual spectrum), a visual seeker cannot lock on.

A NN's power is that is a direct method; it evaluates how close a pattern is to one it was trained upon. Thus, noisy garbage is rejected because it simply bears no resemblance to a desired target. Also, since a van resembles an APC, the NN would, and did, also select it as a likely target. This double-barrel power, adaptive target ID along with imperviousness to noise, compounded with the sensitivity of TT to noise, clearly shows the superiority of NNs in target ID

ACKNOWLEDGMENTS

I would like to thank the following people for making this summer research possible:

(AGI/KISS)

Mike Wallace, for being such a cool mentor
Michael Deiler, for his guidance in my pursuit of an Amiga
Lee Prestwood, for possibly being more lax than me
Thomas Marler, for relaying to me the beauty of Cozumel
Neville Thomson and Russel Dukes, for their timely bits of wisdom

SPACE DEBRIS ANALYSIS

By
Dana Farver

Mentor: David M. Hogg
Office: AFATL / SAA
Date: 15 Aug. 1990

A special thanks to

David M. Hogg

Mr. Don Harrison

Lt Col Donohue

C1C Charlie Schlegel

C1C Rob Haug

Mr. Charlie Miller

and

All of SAA

Background

Earth's orbital environment has become a space junkyard. The dangers of space debris, however, do not come from natural bodies such as meteoroids, asteroids, or the ejects from comets. There is a growing concern with man-made or artificial space debris such as dead satellites, spent rocket stages, discarded equipment, or fragments from satellite breakups. Of all orbital satellites less than 5% are operational and since the early 1950's the known satellite population has been increasing at a rate of 270 satellites annually.

At face value the increase in satellites would not seem to be a problem since most missions are short-lived and reenter the Earth's atmosphere. The North American Aerospace Defense Command (NORAD) reports that man-made objects reenter at the rate of about one a day. The source of the problem, however, is (1) normal satellite operations, (2) the deterioration of satellites, and (3) the fragmentation or breakup of satellites. Satellite fragmentations are responsible for more than one-third of all known Earth satellites now in orbit.

Debris can not only impair the functioning of a satellite, but it may also create economic consequences. Collisions with small particles (less than 1 mm diameters) usually do not cause catastrophic damage, but they are much more common. The result is the gradual degradation of highly machined surfaces that necessitate expensive replacement. Collisions with large space debris can result in damage exceeding that of explosives. Also the collision is likely to produce millions of smaller particles

4.78	1	0.000526	8.7E-07
4.79	9	0.000527	8.71E-07
4.83	15	0.000528	8.76E-07
4.86	1	0.000529	8.79E-07
4.87	6	0.000529	8.81E-07
4.96	18	0.000533	8.92E-07
5.06	36	0.000536	9.03E-07
5.1	1	0.000538	9.08E-07
5.15	9	0.000539	9.14E-07
5.25	9	0.000543	9.26E-07
5.31	2	0.000545	9.33E-07
5.42	4	0.000549	9.46E-07
5.46	51	0.00055	9.5E-07
5.56	2	0.000553	9.62E-07
5.57	10	0.000554	9.63E-07
5.68	1	0.000557	9.76E-07
5.69	55	0.000558	9.77E-07
6.65	1	0.000587	1.08E-06
6.65	6	0.000587	1.08E-06
6.81	46	0.000592	1.1E-06
6.93	4	0.000596	1.11E-06
7.02	11	0.000598	1.12E-06
7.04	23	0.000599	1.13E-06
7.04	10	0.000599	1.13E-06
7.05	14	0.000599	1.13E-06
7.2	10	0.000603	1.14E-06
7.35	1	0.000607	1.16E-06
7.63	9	0.000615	1.19E-06
7.75	10	0.000618	1.2E-06
7.77	5	0.000619	1.2E-06
8.21	11	0.00063	1.25E-06
8.21	1	0.00063	1.25E-06
8.28	15	0.000632	1.25E-06
8.29	4	0.000632	1.26E-06
8.9	20.5	0.000647	1.32E-06
9	13	0.00065	1.33E-06
9.11	8	0.000652	1.34E-06
9.14	1	0.000653	1.34E-06
9.2	17.5	0.000655	1.35E-06
9.39	20	0.000659	1.36E-06
9.45	8	0.00066	1.37E-06
9.57	1	0.000663	1.38E-06
9.91	4.5	0.000671	1.41E-06
9.96	26	0.000672	1.42E-06
10.14	10.5	0.000676	1.44E-06
10.25	6	0.000679	1.45E-06
10.25	5	0.000679	1.45E-06
10.28	3	0.000679	1.45E-06
10.61	2	0.000686	1.48E-06
10.71	11.5	0.000689	1.49E-06
10.72	3	0.000689	1.49E-06
10.75	8	0.000689	1.49E-06
10.84	19	0.000691	1.5E-06
10.93	1	0.000693	1.51E-06
11.23	10	0.000699	1.54E-06
11.42	9	0.000703	1.55E-06

7.860427	2488.554	147.6693
35.10688	2490.288	2951.82
49.71598	2497.201	5969.105
7.860427	2502.36	150.1408
26.6351	2504.075	1727.458
56.28949	2519.407	7857.897
90.25317	2536.226	20608.46
7.860427	2542.891	157.5551
35.10688	2551.175	3173.669
35.10688	2567.581	3235.293
12.60321	2577.325	421.7229
20.20768	2595.001	1106.629
114.4176	2601.369	35739.47
12.60321	2617.154	441.578
37.71883	2618.723	3962.247
7.860427	2635.849	175.4731
120.4559	2637.395	41279.88
7.860427	2778.081	205.4395
26.6351	2778.081	2358.85
106.6524	2800.185	38730.94
20.20768	2816.537	1414.934
40.24864	2828.679	5686.035
66.51741	2831.361	15574.47
37.71883	2831.361	5007.939
47.43378	2832.701	7931.121
37.71883	2852.651	5121.755
7.860427	2872.325	227.0647
35.10688	2908.345	4701.96
37.71883	2923.513	5513.001
23.52464	2926.025	2149.993
40.24864	2980.246	6649.907
7.860427	2980.246	253.6328
49.71598	2988.692	10232.75
20.20768	2989.895	1692.612
61.50316	3061.501	16832.74
45.09894	3072.925	9152.615
32.40049	3085.393	4781.802
7.860427	3088.777	282.3635
55.21973	3095.521	14026.41
60.47743	3116.685	17172.05
32.40049	3123.31	4960.266
7.860427	3136.474	295.6475
21.89561	3173.187	2375.515
72.31052	3178.515	26039.48
38.99334	3197.548	7708.837
26.6351	3209.069	3635.621
23.52464	3209.069	2834.22
35.10688	3210.197	6335.013
7.860427	3246.207	327.7764
41.48587	3256.374	9216.369
16.61189	3257.387	1479.119
32.40049	3260.423	5642.631
58.40104	3269.496	18485.89
7.860427	3278.52	337.6622
37.71883	3308.245	7988.516
35.10688	3326.798	7037.534

Procedure

With the increase in man-made debris there came a growing concern with collisions caused by this debris. In order to characterize the fragmentation produced by a hypervelocity impact in space, this project was originated. It was created as a "piggy-back" project. The project was added onto an existing test that already had a hypervelocity impact, but the primary objective of the test was not the same as that of the project, to gather fragmentation data.

In the test a small mass was projected at a target. After impact the fragments were collected in styrofoam ejecta acquisition tiles (SEATs). Five different SEATs were placed within the chamber (See figures A-1 and A-2). Each SEAT was broken into four 6/6-inch tiles labeled A through D with different thicknesses (See figure A-3). Tiles A and C were made of Celotex, tiles B and D were styrofoam. The Celotex proved to be too dense to easily extract the fragments; therefore, tiles B and D were the only parts used in the analysis. The purpose of the different parts was to determine the best material and thickness needed to catch the fragments.

The formula used to calculate the velocities of the fragments was obtained from research done by the U.S. Army at the Ballistic Research Laboratory. The report, Technical Report BRL-TR-3054, derives four equations which relate the depth of penetration of the fragment to the fragment's velocity. Of the four equations, one is considered the most accurate:

$$\text{Velocity} = \left(\frac{\rho \, c \, A \, P}{\beta \, M} \right)^{\beta} \qquad (1)$$

where

ρ Density of the impact medium (SEATs) = 31.04 kg/m^3

c Constant determined by Equation 2. (m/s)$^{1/B}$

A Cross sectional area of the fragment, m^2

P Depth of penetration of fragment into impact medium, m

β Constant dependent on the impact medium, (dimensionless)

M Mass of the fragment, kg

$$c = \frac{4}{3} \frac{\rho_s}{\rho} R \beta e^{(Bsub1/\beta)} \qquad (2)$$

where

ρ_s Density of the fragment = 7833 kg/m^3

R Radius of the fragment, m

B$_1$ Constant dependent on the impact medium, (dimensionless)

Since the fragment shape is assumed to be spherical the equation can be derived for the cross sectional area:

$$\text{Area} = \pi \left(\frac{3 \, m}{4 \pi \, \rho_s} \right)^{2/3} \qquad (3)$$

The constants β and B_1 are determined by calibrating the SEATs. To do this it is necessary to measure the velocity, penetration, and mass of the projectile. Steel BBs were used as the projectiles. The mass of the BBs was 349 mg and the density was known to be 7833 kg/m^3. With the help of an oscilloscope the time was recorded between the gun firing and impact. The distance from the gun to the SEAT was also measured. These two figures calculated the needed velocity. The depth of penetration was recorded as it was necessary to plot the relationship. Both constants could then be taken from the graph of the natural log of penetration versus the natural log of velocity. Using linear regression three lines were drawn that best fit the data. Beta was determined from the slope of the line and B_1 from the ordinate intercept. This created a range of values for the two constants. The mass and depth of penetration were then substituted into Equation 1 for each set of constants to get a predicted velocity for each BB. The three predicted velocities were compared to the measured velocity. The constants 0.68 and 0.77 best fit the measured velocities.

With the values for Beta and B_1, it was possible to begin to calculate the velocities of the fragments. In order to do this the data of the mass and penetration had to be gathered. This was done by first dissecting a SEAT until a fragment was found. Then the depth was measured and the angle of entry was taken into account. If the direction of the fragment could not be determined then the perpendicular distance to the front surface was used instead. Later the fragment would be weighed using a mi-

crescale sensitive to the hundred-thousandth of a gram. The mass and penetration data was entered into a spreadsheet with the formulas for the radius, area, velocity, c, and kinetic energy (See Appendix B). From this information a graph could be plotted showing the mass versus the velocity. The regression output determined a line that best fit the data. The constant was the y-intercept and the x coefficient was the slope of the line. The line resulting from the regression analysis was plotted with the existing relationship of mass and velocity (See Appendix C). These graphs were analyzed and conclusions were drawn.

Results and Conclusions

Once the graphs were drawn an analysis could be performed. In each of the shots the regression output showed a positive x-coefficient. The graph of the line was almost flat. It seems to prove the hypothesis that the fragments were riding on a shock wave from the impact; therefore, they were traveling at a constant velocity. At the heavier masses the line had an upward trend. This follows the belief that the heavier masses have more momentum and kinetic energy and as a result will penetrate the medium further.

Bibliography

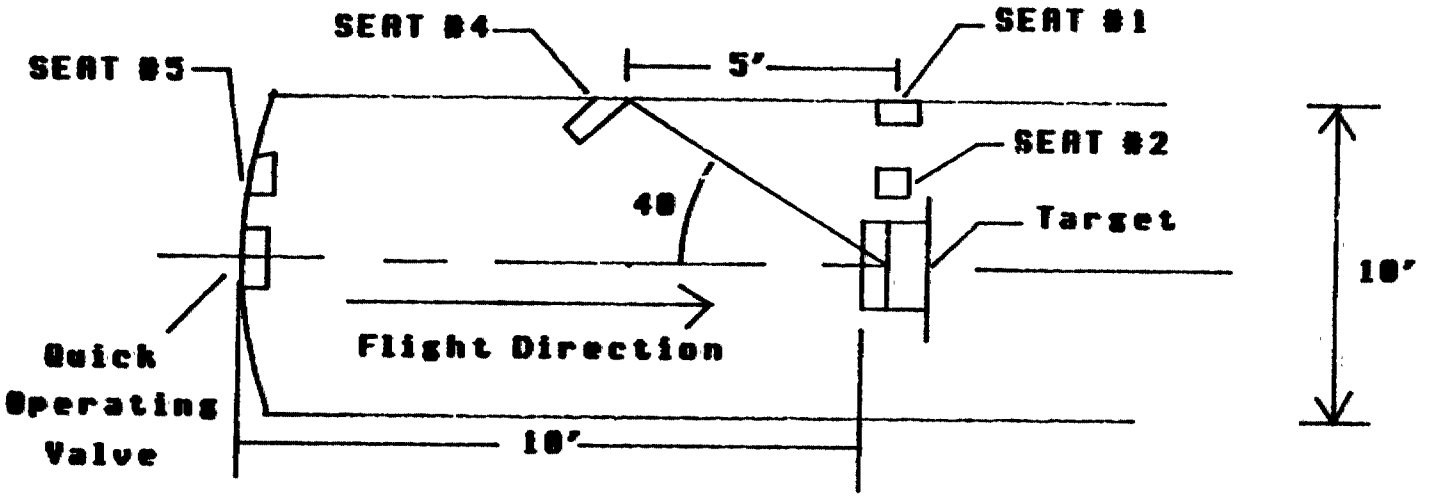
Grace, Fred I. and Hubert W. Meyer, Jr. "The Penetration of Fragments into Collection Media". Technical Report BRL-TR-3054. Ballistic Research Laboratory, MD: 1989.

Johnson, Nicholas L. and Darren S. McKnight. Artificial Space Debris. Orbit Book Company, Malabar, FL: 1987.

Schlegel, Charles F. and Robert L. Haug. "Hypervelocity Impact Analysis". US Air Force Academy: May 1990.

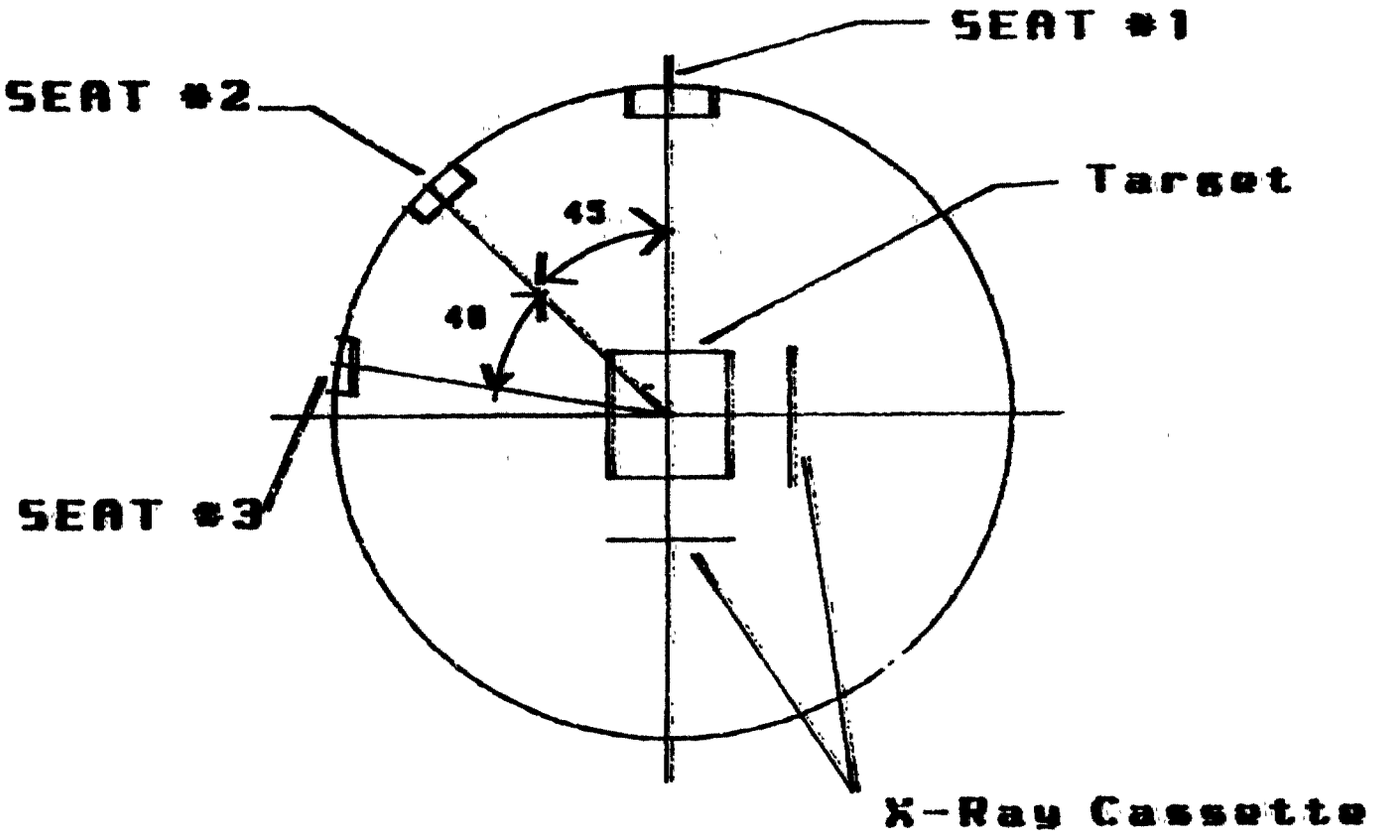
APPENDIX A

SIDE VIEW

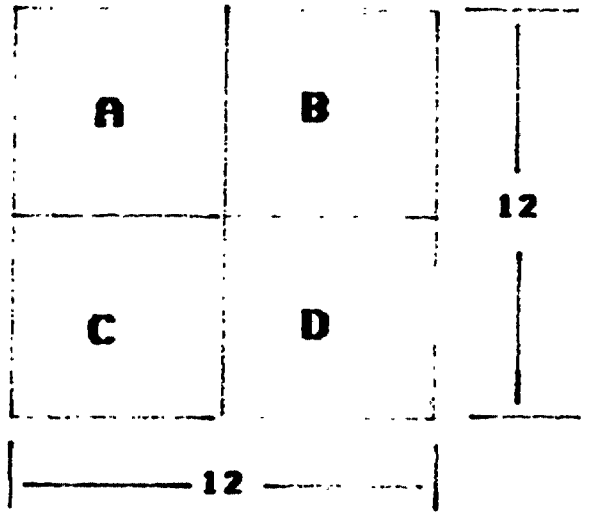


= All Dimensions in Feet

DOWNRANGE VIEW



SEAT Configuration



A - 6 X 6 X 3/4 Celotex
B - 6 X 6 X 1 Styrofoam
C - 6 X 6 X 3/4 Celotex
D - 6 X 6 X 3 Styrofoam

APPENDIX B

SHOT 6575

Rho = 31.0426
Rhosubs = 7833
Beta =0.681112
Bsubl =6.766796

Mass	Penetration	r	a	Calc Velocity	c	Kinetic Energy
0.18	3	0.000176	9.77E-08	16.61189	834.1201	24.83595
0.22	3	0.000189	1.12E-07	16.61189	891.823	30.35506
0.25	4	0.000197	1.22E-07	20.20768	930.6458	51.04379
0.25	3	0.000197	1.22E-07	16.61189	930.6458	34.49438
0.26	6	0.000199	1.25E-07	26.6351	942.8926	92.22571
0.28	6	0.000204	1.31E-07	26.6351	966.4746	96.31999
0.29	3	0.000207	1.34E-07	16.61189	977.8459	40.01348
0.31	2.5	0.000211	1.4E-07	14.67195	999.8273	33.36625
0.31	3	0.000211	1.4E-07	16.61189	999.8273	42.77303
0.34	9	0.000218	1.49E-07	35.10688	1031.092	209.5238
0.34	1	0.000218	1.49E-07	7.860427	1031.092	10.50367
0.37	3.5	0.000224	1.58E-07	18.45089	1060.568	62.98057
0.39	4	0.000228	1.64E-07	20.20768	1079.343	79.62831
0.4	8	0.00023	1.66E-07	32.40049	1088.49	209.9584
0.41	4	0.000232	1.69E-07	20.20768	1097.486	83.71181
0.41	3	0.000232	1.69E-07	16.61189	1097.486	56.57079
0.42	10	0.000234	1.72E-07	37.71883	1106.337	298.7691
0.43	4.5	0.000236	1.75E-07	21.89561	1115.049	103.0748
0.44	1	0.000238	1.77E-07	7.860427	1123.627	13.59299
0.46	5	0.000241	1.83E-07	23.52464	1140.4	127.284
0.46	2	0.000241	1.83E-07	12.60321	1140.4	36.53343
0.47	9	0.000243	1.85E-07	35.10688	1148.604	289.6358
0.48	4	0.000245	1.88E-07	20.20768	1156.693	98.00408
0.52	1	0.000251	1.98E-07	7.860427	1187.97	16.06444
0.54	6	0.000254	2.03E-07	26.6351	1203.009	191.5457
0.54	1	0.000254	2.03E-07	7.860427	1203.009	16.68231
0.54	4	0.000254	2.03E-07	20.20768	1203.009	110.2546
0.55	3	0.000256	2.06E-07	16.61189	1210.39	75.88764
0.55	9	0.000256	2.06E-07	35.10688	1210.39	338.9355
0.55	7	0.000256	2.06E-07	29.58371	1210.39	240.6788
0.58	1	0.00026	2.13E-07	7.860427	1232.009	17.91803
0.59	1.5	0.000262	2.16E-07	10.36058	1239.049	31.66577
0.59	1	0.000262	2.16E-07	7.860427	1239.049	18.22696
0.59	9	0.000262	2.16E-07	35.10688	1239.049	363.5854
0.6	25	0.000263	2.18E-07	70.40441	1246.01	1487.034
0.63	3	0.000268	2.25E-07	16.61189	1266.44	86.92584
0.65	3	0.000271	2.3E-07	16.61189	1279.702	89.68539
0.66	2	0.000272	2.32E-07	12.60321	1286.231	52.41753
0.66	2	0.000272	2.32E-07	12.60321	1286.231	52.41753
0.68	1	0.000275	2.37E-07	7.860427	1299.094	21.00735
0.69	2	0.000276	2.39E-07	12.60321	1305.432	54.80014
0.69	11.5	0.000276	2.39E-07	41.48587	1305.432	593.7717
0.72	9	0.00028	2.46E-07	35.10688	1324.083	443.6974
0.74	12.5	0.000283	2.51E-07	43.91013	1336.231	713.3969
0.76	3	0.000285	2.55E-07	16.61189	1348.163	104.8629
0.76	5	0.000285	2.55E-07	23.52464	1348.163	210.2953
0.76	11	0.000285	2.55E-07	40.24864	1348.163	615.5822
0.77	12	0.000286	2.58E-07	42.70606	1354.05	702.1658
0.77	10	0.000286	2.58E-07	37.71883	1354.05	547.7433
0.78	5	0.000288	2.6E-07	23.52464	1359.886	215.8294
0.79	20	0.000289	2.62E-07	60.47743	1365.673	1444.72
0.8	8	0.00029	2.64E-07	32.40049	1371.411	419.9167

0.81	12	0.000291	2.86E-07
0.86	6	0.000297	2.77E-07
0.86	4	0.000297	2.77E-07
0.87	6	0.000298	2.79E-07
0.88	12	0.000299	2.81E-07
0.88	5	0.000299	2.81E-07
0.92	8	0.000304	2.9E-07
0.92	4	0.000304	2.9E-07
0.93	3	0.000305	2.92E-07
0.94	6	0.000306	2.94E-07
0.95	10.5	0.000307	2.96E-07
0.95	2.5	0.000307	2.96E-07
0.95	1	0.000307	2.96E-07
0.95	3	0.000307	2.96E-07
0.96	26	0.000308	2.98E-07
0.96	2	0.000308	2.98E-07
0.97	9	0.000309	3E-07
0.98	3	0.00031	3.02E-07
0.99	7	0.000311	3.04E-07
0.99	2.5	0.000311	3.04E-07
0.99	3	0.000311	3.04E-07
1	13	0.000312	3.07E-07
1.01	4	0.000313	3.09E-07
1.02	14	0.000314	3.11E-07
1.02	6	0.000314	3.11E-07
1.02	4	0.000314	3.11E-07
1.03	5	0.000315	3.13E-07
1.04	5	0.000316	3.15E-07
1.04	2.5	0.000316	3.15E-07
1.07	6	0.000319	3.21E-07
1.07	6.5	0.000319	3.21E-07
1.08	7	0.00032	3.23E-07
1.08	1	0.00032	3.23E-07
1.12	8	0.000324	3.31E-07
1.13	11	0.000325	3.33E-07
1.14	5.5	0.000326	3.35E-07
1.15	15	0.000327	3.36E-07
1.16	9	0.000328	3.38E-07
1.17	8	0.000329	3.4E-07
1.18	4	0.00033	3.42E-07
1.19	5	0.000331	3.44E-07
1.2	6	0.000332	3.46E-07
1.2	17	0.000332	3.46E-07
1.21	3.5	0.000333	3.48E-07
1.22	6	0.000334	3.5E-07
1.24	3.5	0.000336	3.54E-07
1.25	9	0.000336	3.56E-07
1.26	8	0.000337	3.58E-07
1.28	6	0.000339	3.61E-07
1.3	5	0.000341	3.65E-07
1.31	1.5	0.000342	3.67E-07
1.31	8	0.000342	3.67E-07
1.33	2	0.000344	3.71E-07
1.33	10	0.000344	3.71E-07
1.35	5	0.000345	3.74E-07
1.38	9	0.000348	3.8E-07

42.70606	1377.102	738.6419
26.6351	1404.874	305.0543
20.20768	1404.874	175.5906
26.6351	1410.298	308.6014
42.70606	1415.681	802.4752
23.52464	1415.681	243.4999
32.40049	1436.813	482.9042
20.20768	1436.813	187.8411
16.61189	1442	128.3191
26.6351	1447.151	333.4314
38.99334	1452.264	722.2283
14.67195	1452.264	102.2514
7.860427	1452.264	29.3485
16.61189	1452.264	131.0786
72.31052	1457.342	2509.829
12.60321	1457.342	76.24368
35.10688	1462.385	597.759
16.61189	1467.393	135.218
29.58371	1472.367	433.2219
14.67195	1472.367	106.5567
16.61189	1472.367	136.5978
45.09894	1477.308	1016.957
20.20768	1482.216	206.2169
47.43378	1487.092	1147.481
26.6351	1487.092	361.8085
20.20768	1487.092	208.2587
23.52464	1491.936	285.0055
23.52464	1496.749	287.7726
14.67195	1496.749	111.9384
26.6351	1511.004	379.5443
28.1275	1511.004	423.2687
29.58371	1515.697	472.6057
7.860427	1515.697	33.36461
32.40049	1534.183	587.8834
40.24864	1538.735	915.2735
25.10245	1543.261	359.1757
49.71598	1547.76	1421.215
35.10688	1552.234	714.8458
32.40049	1556.681	614.1282
20.20768	1561.104	240.9267
23.52464	1565.501	329.2782
26.6351	1569.874	425.6571
54.14018	1569.874	1758.695
18.45089	1574.223	205.9635
26.6351	1578.548	432.7514
18.45089	1587.127	211.07
35.10688	1591.382	770.308
32.40049	1595.614	661.3688
26.6351	1604.013	454.0342
23.52464	1612.324	359.7157
10.36058	1616.447	70.30875
32.40049	1616.447	687.0136
12.60321	1624.632	105.6
37.71883	1624.632	946.1
23.52464	1632.735	373.55
35.10688	1644.741	850.42

1.39	2	0.000349	3.82E-07
1.43	17.5	0.000352	3.89E-07
1.47	2	0.000355	3.96E-07
1.48	12	0.000356	3.98E-07
1.49	9	0.000357	4E-07
1.5	11	0.000358	4.02E-07
1.5	17	0.000358	4.02E-07
1.52	4	0.000359	4.05E-07
1.54	5	0.000361	4.09E-07
1.6	1	0.000365	4.19E-07
1.6	3	0.000365	4.19E-07
1.61	5	0.000366	4.21E-07
1.66	8	0.00037	4.3E-07
1.66	1	0.00037	4.3E-07
1.66	25	0.00037	4.3E-07
1.69	11	0.000372	4.35E-07
1.7	9	0.000373	4.37E-07
1.71	4	0.000374	4.38E-07
1.73	2	0.000375	4.42E-07
1.74	12	0.000376	4.43E-07
1.77	11	0.000378	4.49E-07
1.77	5	0.000378	4.49E-07
1.78	1	0.000379	4.5E-07
1.78	29	0.000379	4.5E-07
1.78	21	0.000379	4.5E-07
1.79	1	0.000379	4.52E-07
1.81	5	0.000381	4.55E-07
1.83	3	0.000382	4.59E-07
1.86	12	0.000384	4.64E-07
1.86	3	0.000384	4.64E-07
1.87	1	0.000385	4.65E-07
1.89	6	0.000386	4.69E-07
1.92	8.5	0.000388	4.74E-07
1.94	22.5	0.00039	4.77E-07
1.95	26	0.00039	4.78E-07
1.95	3	0.00039	4.78E-07
1.98	4	0.000392	4.83E-07
1.99	5	0.000393	4.85E-07
2.04	1	0.000396	4.93E-07
2.06	32.5	0.000397	4.96E-07
2.1	14	0.0004	5.03E-07
2.13	2	0.000402	5.07E-07
2.21	5	0.000407	5.2E-07
2.23	5.5	0.000408	5.23E-07
2.28	6	0.000411	5.31E-07
2.28	6.5	0.000411	5.31E-07
2.31	1	0.000413	5.36E-07
2.31	9	0.000413	5.36E-07
2.32	3	0.000414	5.37E-07
2.34	2	0.000415	5.4E-07
2.35	9	0.000415	5.42E-07
2.37	7.5	0.000416	5.45E-07
2.44	10	0.000421	5.56E-07
2.45	2.5	0.000421	5.57E-07
2.48	22.5	0.000423	5.62E-07
2.51	3	0.000425	5.66E-07

12.60321	16.8.704	110.3945
55.21973	1864.37	2180.191
12.60321	1679.746	116.7481
42.70606	1683.546	1349.617
35.10688	1687.329	918.2071
40.24864	1691.096	1214.965
54.14018	1691.096	2198.369
20.20768	1698.579	310.3462
23.52464	1705.996	426.1248
7.860427	1727.87	49.42906
16.61189	1727.87	220.764
23.52464	1731.462	445.4941
32.40049	1749.204	871.3272
7.860427	1749.204	51.28265
70.40441	1749.204	4114.128
40.24864	1759.679	1368.86
35.10688	1763.142	1047.619
20.20768	1766.593	349.1395
12.60321	1773.453	137.3975
42.70606	1776.864	1586.712
40.24864	1787.018	1433.658
23.52464	1787.018	489.7668
7.860427	1790.377	54.98982
77.89382	1790.377	5400.028
62.52095	1790.377	3478.894
7.860427	1793.723	55.29876
23.52464	1800.379	500.8349
16.61189	1806.986	252.4989
42.70606	1816.807	1696.141
16.61189	1816.807	256.6382
7.860427	1820.057	57.77021
26.6351	1826.523	670.4099
33.76638	1836.136	1094.562
65.52905	1842.489	4165.235
72.31052	1845.65	5098.091
16.61189	1845.65	269.0562
20.20768	1855.067	404.2668
23.52464	1858.184	550.6417
7.860427	1873.618	63.02205
84.17983	1879.721	7298.832
47.43378	1891.81	2362.462
12.60321	1900.776	169.1657
23.52464	1924.281	611.5167
25.10245	1930.068	702.5982
26.6351	1944.387	808.7485
28.1275	1944.387	901.9184
7.860427	1952.878	71.3632
35.10688	1952.878	1423.529
16.61189	1955.692	320.1079
12.60321	1961.296	185.844
35.10688	1964.085	1448.179
31.00708	1969.642	1139.305
37.71883	1988.845	1735.706
14.67195	1991.559	263.701
65.52905	1999.655	5324.631
16.61189	2007.685	346.3236

2.52	14.5	0.000425	5.68E-07
2.54	6.5	0.000426	5.71E-07
2.55	11	0.000427	5.72E-07
2.59	8	0.000429	5.78E-07
2.6	6	0.00043	5.8E-07
2.61	14	0.00043	5.81E-07
2.62	12	0.000431	5.83E-07
2.66	4	0.000433	5.88E-07
2.77	1	0.000439	6.05E-07
2.82	4	0.000441	6.12E-07
2.82	9	0.000441	6.12E-07
2.86	8	0.000443	6.18E-07
2.87	1	0.000444	6.19E-07
2.87	12	0.000444	6.19E-07
2.89	9	0.000445	6.22E-07
2.94	5	0.000447	6.29E-07
2.96	6	0.000448	6.32E-07
2.98	37	0.00045	6.35E-07
2.99	6.5	0.00045	6.36E-07
3.04	9	0.000453	6.43E-07
3.07	3	0.000454	6.47E-07
3.1	3.5	0.000455	6.52E-07
3.1	1	0.000455	6.52E-07
3.21	20	0.000461	6.67E-07
3.25	2	0.000463	6.73E-07
3.33	2	0.000466	6.84E-07
3.34	24.5	0.000467	6.85E-07
3.36	4.5	0.000468	6.88E-07
3.42	9.5	0.000471	6.96E-07
3.45	1	0.000472	7E-07
3.45	5	0.000472	7E-07
3.54	16	0.000476	7.12E-07
3.57	7	0.000477	7.16E-07
3.58	22	0.000478	7.17E-07
3.63	1.5	0.00048	7.24E-07
3.79	4.5	0.000487	7.45E-07
3.81	8.5	0.000488	7.48E-07
3.82	14	0.000488	7.49E-07
3.84	1	0.000489	7.52E-07
3.85	7.5	0.00049	7.53E-07
4	5	0.000496	7.72E-07
4.06	18	0.000498	7.8E-07
4.1	12	0.0005	7.85E-07
4.15	14	0.000502	7.92E-07
4.21	7	0.000504	7.99E-07
4.28	8	0.000507	8.08E-07
4.3	29	0.000508	8.11E-07
4.33	2.5	0.000509	8.14E-07
4.43	4	0.000513	8.27E-07
4.45	2.5	0.000514	8.29E-07
4.54	10	0.000517	8.4E-07
4.59	17	0.000519	8.47E-07
4.62	11	0.00052	8.5E-07
4.63	3	0.000521	8.52E-07
4.65	22	0.000521	8.54E-07
4.74	1	0.000525	8.65E-07

48.58116	2010.348	2973.762
28.1275	2015.653	1004.769
40.24864	2018.294	2065.44
32.40049	2028.793	1359.48
26.6351	2031.4	922.2571
47.43378	2034.001	2936.202
42.70606	2036.596	2389.187
20.20768	2046.908	543.1059
7.860427	2074.743	85.57405
20.20768	2087.152	575.7739
35.10688	2087.152	1737.815
32.40049	2096.974	1501.202
7.860427	2099.415	88.66337
42.70606	2099.415	2617.163
35.10688	2104.281	1780.952
23.52464	2116.347	813.5109
26.6351	2121.135	1049.954
91.95327	2125.902	12598.55
28.1275	2128.277	1182.779
32.40049	2140.075	1595.684
16.61189	2147.091	423.591
18.45089	2154.063	527.6751
7.860427	2154.063	95.7688
60.47743	2179.245	5870.318
12.60321	2188.26	258.1166
12.60321	2206.069	264.4703
69.44226	2208.275	8053.12
21.89561	2212.674	805.4218
36.42382	2225.767	2268.647
7.860427	2232.257	106.5814
23.52464	2232.257	954.6301
51.95014	2251.501	4776.906
29.58371	2257.843	1562.224
64.53367	2259.95	7454.624
10.36058	2270.422	194.825
21.89561	2303.302	908.4966
33.76638	2307.346	2172.021
47.43378	2309.363	4297.43
7.860427	2313.386	118.6297
31.00708	2315.393	1850.77
23.52464	2345.081	1106.818
56.28949	2356.748	6432.069
42.70606	2364.462	3738.805
47.43378	2374.035	4668.674
29.58371	2385.421	1842.287
32.40049	2398.57	2246.554
77.89382	2402.3	13045.01
14.67195	2407.874	466.0512
20.20768	2426.269	904.4959
14.67195	2429.915	478.9672
37.71883	2446.187	3229.551
54.14018	2455.135	6727.01
40.24864	2460.472	3742.092
16.61189	2462.246	638.8359
64.53367	2465.786	9682.682
7.860427	2481.593	146.4336

4.78	1	0.000526	8.7E-07	7.860427	2488.554	147.6693
4.79	9	0.000527	8.71E-07	35.10688	2490.288	2951.82
4.83	15	0.000528	8.76E-07	49.71598	2497.201	5969.105
4.86	1	0.000529	8.79E-07	7.860427	2502.36	150.1408
4.87	6	0.000529	8.81E-07	26.6351	2504.075	1727.458
4.96	18	0.000533	8.92E-07	56.28949	2519.407	7857.897
5.06	36	0.000536	9.03E-07	90.25317	2536.226	20608.46
5.1	1	0.000538	9.08E-07	7.860427	2542.891	157.5551
5.15	9	0.000539	9.14E-07	35.10688	2551.175	3173.669
5.25	9	0.000543	9.26E-07	35.10688	2567.581	3235.293
5.31	2	0.000545	9.33E-07	12.60321	2577.325	421.7229
5.42	4	0.000549	9.46E-07	20.20768	2595.001	1106.629
5.46	51	0.00055	9.5E-07	114.4176	2601.369	35739.47
5.56	2	0.000553	9.62E-07	12.60321	2617.154	441.578
5.57	10	0.000554	9.63E-07	37.71883	2618.723	3962.247
5.68	1	0.000557	9.76E-07	7.860427	2635.849	175.4731
5.69	55	0.000558	9.77E-07	120.4559	2637.395	41279.88
6.65	1	0.000587	1.08E-06	7.860427	2778.081	205.4395
6.65	6	0.000587	1.08E-06	26.6351	2778.081	2358.85
6.81	46	0.000592	1.1E-06	106.6524	2800.185	38730.94
6.93	4	0.000596	1.11E-06	20.20768	2816.537	1414.934
7.02	11	0.000598	1.12E-06	40.24864	2828.679	5686.035
7.04	23	0.000599	1.13E-06	66.51741	2831.361	15574.47
7.04	10	0.000599	1.13E-06	37.71883	2831.361	5007.939
7.05	14	0.000599	1.13E-06	47.43378	2832.701	7931.121
7.2	10	0.000603	1.14E-06	37.71883	2852.651	5121.755
7.35	1	0.000607	1.16E-06	7.860427	2872.325	227.0647
7.63	9	0.000615	1.19E-06	35.10688	2908.345	4701.96
7.75	10	0.000618	1.2E-06	37.71883	2923.513	5513.001
7.77	5	0.000619	1.2E-06	23.52464	2926.025	2149.993
8.21	11	0.00063	1.25E-06	40.24864	2980.246	6649.907
8.21	1	0.00063	1.25E-06	7.860427	2980.246	253.6328
8.28	15	0.000632	1.25E-06	49.71598	2988.692	10232.75
8.29	4	0.000632	1.26E-06	20.20768	2989.895	1692.612
8.9	20.5	0.000647	1.32E-06	61.50316	3061.501	16832.74
9	13	0.00065	1.33E-06	45.09894	3072.925	9152.615
9.11	8	0.000652	1.34E-06	32.40049	3085.393	4781.802
9.14	1	0.000653	1.34E-06	7.860427	3088.777	282.3635
9.2	17.5	0.000655	1.35E-06	55.21973	3095.521	14026.41
9.39	20	0.000659	1.36E-06	60.47743	3116.685	17172.05
9.45	8	0.00066	1.37E-06	32.40049	3123.31	4960.266
9.57	1	0.000663	1.38E-06	7.860427	3136.474	295.6475
9.91	4.5	0.000671	1.41E-06	21.89561	3173.187	2375.515
9.96	26	0.000672	1.42E-06	72.31052	3178.515	26039.48
10.14	10.5	0.000676	1.44E-06	38.99334	3197.548	7708.837
10.25	6	0.000679	1.45E-06	26.6351	3209.069	3635.921
10.25	5	0.000679	1.45E-06	23.52464	3209.069	3033.22
10.28	3	0.000679	1.45E-06	35.10688	3212.197	6335.013
10.61	1	0.000686	1.48E-06	7.860427	3246.207	327.7764
10.71	11.5	0.000689	1.49E-06	41.48587	3256.374	9216.369
10.72	3	0.000689	1.49E-06	16.61189	3257.387	1479.119
10.75	8	0.000689	1.49E-06	32.40049	3260.423	5642.631
10.84	19	0.000691	1.5E-06	58.40104	3269.496	18485.89
10.93	1	0.000693	1.51E-06	7.860427	3278.52	337.6622
11.23	10	0.000699	1.54E-06	37.71883	3308.245	7988.516
11.42	9	0.000703	1.55E-06	35.10688	3326.798	7037.534

11.43	11	0.000704	1.56E-06	40.24864	3327.769	9258.032
11.6	4	0.000707	1.57E-06	20.20768	3344.186	2368.432
11.68	14	0.000709	1.58E-06	47.43378	3351.856	13139.79
12.02	1	0.000716	1.61E-06	7.860427	3384.069	371.3358
12.46	31	0.000724	1.65E-06	81.51368	3424.868	41395.11
12.98	63.5	0.000734	1.69E-06	132.8425	3471.864	114529.8
13.23	6.5	0.000739	1.71E-06	28.1275	3494.012	5233.5
13.36	1	0.000741	1.73E-06	7.860427	3505.419	412.7326
13.41	21	0.000742	1.73E-06	62.52095	3509.786	26208.97
13.55	1.5	0.000745	1.74E-06	10.36058	3521.958	727.2393
13.92	19	0.000751	1.77E-06	58.40104	3553.728	23738.34
14.25	14.5	0.000757	1.8E-06	48.58116	3581.591	16815.92
14.39	1	0.00076	1.81E-06	7.860427	3593.282	444.5526
14.55	17	0.000763	1.83E-06	54.14018	3606.551	21324.18
14.86	10	0.000768	1.85E-06	37.71883	3631.985	10570.73
14.89	26	0.000768	1.86E-06	72.31052	3634.427	38928.5
15.14	2	0.000773	1.88E-06	12.60321	3654.655	1202.426
15.23	10	0.000774	1.88E-06	37.71883	3661.882	10833.94
17.98	10.5	0.000818	2.1E-06	38.99334	3870.208	13669.12
19.34	56	0.000838	2.21E-06	121.9433	3965.427	143794.6
20.05	20	0.000849	2.26E-06	60.47743	4013.37	36666.63
20.13	26	0.00085	2.27E-06	72.31052	4018.701	52627.99
20.18	15	0.00085	2.27E-06	49.71598	4022.025	24939.24
20.35	1	0.000853	2.28E-06	7.860427	4033.288	628.6758
21.95	11	0.000875	2.4E-06	40.24864	4136.337	17778.99
22.86	15	0.000887	2.47E-06	49.71598	4192.726	28251.29
23.49	5	0.000895	2.51E-06	23.52464	4230.894	6499.786
24.61	23.5	0.000909	2.59E-06	67.49894	4297.095	56062.89
26.3	21	0.000929	2.71E-06	62.52095	4393.288	51401.64
26.56	26	0.000932	2.73E-06	72.31052	4407.718	69438.62
27.16	8	0.000939	2.77E-06	32.40049	4440.662	14256.17
27.5	13.5	0.000943	2.79E-06	46.27326	4459.115	29441.7
28.77	16	0.000957	2.88E-06	51.95014	4526.728	38822.49
29.4	25	0.000964	2.92E-06	70.40441	4559.531	72864.68
30.34	15	0.000974	2.98E-06	49.71598	4607.616	37495.37
30.5	10	0.000976	2.99E-06	37.71883	4615.701	21696.32
31	1.5	0.000981	3.02E-06	10.36058	4640.787	1663.795
31.58	11	0.000987	3.06E-06	40.24864	4669.551	25579.06
32.78	22	0.001	3.14E-06	64.53367	4727.963	68257.7
35.15	22	0.001023	3.29E-06	64.53367	4839.266	73192.75
35.55	37	0.001027	3.31E-06	91.95327	4857.554	150294.8
35.73	25	0.001029	3.33E-06	70.40441	4865.739	88552.9
36.41	3	0.001035	3.37E-06	16.61189	4896.412	5023.762
37.25	25	0.001043	3.42E-06	70.40441	4933.781	92320.05
37.41	15	0.001045	3.43E-06	49.71598	4940.935	46232.76
39.15	17.5	0.001061	3.53E-06	55.21973	5016.279	59688.46
40.62	51	0.001074	3.62E-06	114.4176	5078.293	265885.9
42.13	18	0.001087	3.71E-06	56.28949	5140.455	66744.6
44.79	22	0.001109	3.87E-06	64.53367	5246.441	93266.09
46.34	1	0.001122	3.95E-06	7.860427	5306.275	1431.589
47.32	13	0.00113	4.01E-06	45.09894	5343.42	48122.42
50.21	17	0.001152	4.17E-06	54.14018	5450.059	73586.75
50.45	19	0.001154	4.19E-06	58.40104	5458.728	86034.44
52.11	1	0.001167	4.28E-06	7.860427	5517.955	1609.843
53.57	14	0.001178	4.36E-06	47.43378	5569.014	60265.27
58.1	17	0.00121	4.6E-06	54.14018	5721.762	85150.17

67.16	7	0.00127	5.06E-06	29.58371	6004.931	29389.07
67.54	14	0.001272	5.08E-06	47.43378	6016.235	75981.26
68.53	1	0.001278	5.13E-06	7.860427	6045.488	2117.108
71.92	1	0.001299	5.3E-06	7.860427	6143.572	2221.836
75.56	22	0.001321	5.48E-06	64.53367	6245.517	157338.4
77.56	27	0.001332	5.57E-06	74.19339	6300.142	213470.7
80.07	14.5	0.001346	5.69E-06	48.58116	6367.384	94487.75
95.97	6	0.00143	6.43E-06	26.6351	6763.678	34041.93
104.42	6	0.001471	6.8E-06	26.6351	6956.631	37039.26
105.36	7	0.001475	6.84E-06	29.58371	6977.444	46105.31
106.52	25	0.001481	6.89E-06	70.40441	7002.957	263998.2
126.21	19	0.001567	7.71E-06	58.40104	7410.298	215231.1
131.11	1	0.001587	7.91E-06	7.860427	7504.983	4050.402
132.53	21	0.001593	7.97E-06	62.52095	7531.98	259021.2
166.32	11	0.001718	9.27E-06	40.24864	8124.299	134715.3
169.6	1	0.001729	9.39E-06	7.860427	8177.359	5239.48
175.53	12	0.001749	9.61E-06	42.70606	8271.575	160066.4
200.3	75	0.001828	1.05E-05	148.7897	8643.668	2217158
204.77	54	0.001841	1.06E-05	118.9598	8707.494	1448895
237.72	10	0.001935	1.18E-05	37.71883	9151.515	169103.3
310.79	7	0.002116	1.41E-05	29.58371	10006.76	136001
314.53	21	0.002124	1.42E-05	62.52095	10046.74	614728.4
321.35	7	0.00214	1.44E-05	29.58371	10118.84	140622.1
355.35	1	0.002212	1.54E-05	7.860427	10463.81	10977.88
359.52	2	0.002221	1.55E-05	12.60321	10504.58	28553.26
368.22	5	0.002239	1.57E-05	23.52464	10588.64	101888.1
439.48	15	0.002375	1.77E-05	49.71598	11231.84	543126.7
452.2	21	0.002398	1.81E-05	62.52095	11339.17	883795.4
723.43	21	0.002804	2.47E-05	62.52095	13261.82	1413897
1330.52	49	0.003436	3.71E-05	111.342	16248.44	8247255
3016.46	74	0.004513	6.4E-05	147.4356	21345.37	32784775
5021.9	64	0.005349	8.99E-05	133.554	25298.44	44787013

SHOT 6579

Rho = 31.04
Rhosubs = 7833
Beta =0.681112
Bsubs =6.766796

Mass	Pene-tration	p	a	Calc Velocity	c	Kinetic Energy
0.12	53.4	0.000154	7.46E-08	118.058	728.7321	836.2608
0.14	4.5	0.000162	8.26E-08	21.89561	767.1556	33.55924
0.21	26.7	0.000186	1.08E-07	73.6309	878.174	569.2585
0.25	5	0.000197	1.22E-07	23.52464	930.7238	69.1761
0.25	1	0.000197	1.22E-07	7.860427	930.7238	7.72329
0.26	3	0.000199	1.25E-07	16.61189	942.9715	35.87416
0.27	5	0.000202	1.28E-07	23.52464	954.9092	74.71018
0.33	3	0.000216	1.46E-07	16.61189	1020.968	45.53258
0.34	11	0.000218	1.49E-07	40.24864	1031.178	275.392
0.34	71.4	0.000218	1.49E-07	143.8872	1031.178	3519.601
0.36	3	0.000222	1.55E-07	16.61189	1051.014	49.67191
0.36	2	0.000222	1.55E-07	12.60321	1051.014	28.59138
0.37	51	0.000224	1.58E-07	114.4176	1060.656	2421.905
0.37	15	0.000224	1.58E-07	49.71598	1060.656	457.2606
0.42	10	0.000234	1.72E-07	37.71883	1106.43	298.7691
0.43	5	0.000236	1.75E-07	23.52464	1115.142	118.9829
0.44	1	0.000238	1.77E-07	7.860427	1123.721	13.59299
0.44	5	0.000238	1.77E-07	23.52464	1123.721	121.7499
0.45	56.4	0.000239	1.8E-07	122.5359	1132.17	3378.386
0.5	13	0.000248	1.93E-07	45.09894	1172.638	508.4786
0.5	11	0.000248	1.93E-07	40.24864	1172.638	404.9883
0.5	6	0.000248	1.93E-07	26.6351	1172.638	177.3571
0.51	13	0.00025	1.96E-07	45.09894	1180.405	518.6482
0.52	1	0.000251	1.98E-07	7.860427	1188.07	16.06444
0.53	50.4	0.000253	2.01E-07	113.499	1195.637	3413.736
0.53	14	0.000253	2.01E-07	47.43378	1195.637	596.2403
0.54	1	0.000254	2.03E-07	7.860427	1203.11	16.68231
0.55	9	0.000256	2.06E-07	35.10688	1210.491	338.9355
0.58	1	0.00026	2.13E-07	7.860427	1232.112	17.91803
0.59	9.5	0.000262	2.16E-07	36.42382	1239.153	391.3748
0.59	1	0.000262	2.16E-07	7.860427	1239.153	18.22696
0.59	1	0.000262	2.16E-07	7.860427	1239.153	18.22696
0.6	25.7	0.000263	2.18E-07	71.74118	1246.114	1544.039
0.6	6	0.000263	2.18E-07	26.6351	1246.114	212.8286
0.61	3	0.000265	2.2E-07	16.61189	1252.999	84.16629
0.62	7	0.000266	2.23E-07	29.58371	1259.809	271.3107
0.62	13	0.000266	2.23E-07	45.09894	1259.809	630.5135
0.64	10	0.000269	2.28E-07	37.71883	1273.212	455.2671
0.65	10.5	0.000271	2.3E-07	38.99334	1279.809	494.1562
0.66	2	0.000272	2.32E-07	12.60321	1286.339	52.41753
0.67	8.5	0.000273	2.35E-07	33.76638	1292.803	381.9565
0.68	14	0.000275	2.37E-07	47.43378	1299.203	764.9875
0.7	2	0.000277	2.42E-07	12.60321	1311.818	55.59435
0.7	12	0.000277	2.42E-07	42.70606	1311.818	638.3325
0.71	1	0.000279	2.44E-07	7.860427	1318.035	21.93414
0.71	1	0.000279	2.44E-07	7.860427	1318.035	21.93414
0.73	20	0.000281	2.49E-07	60.47743	1330.296	1334.994

0.74	50.4	0.000283	2.51E-07	213.499	1336.343	4766.348
0.74	7	0.000283	2.51E-07	29.58371	1336.343	323.8224
0.76	5	0.000285	2.55E-07	23.52464	1348.276	210.2953
0.76	3.5	0.000285	2.55E-07	18.45089	1348.276	129.3655
0.76	13	0.000285	2.55E-07	45.09894	1348.276	772.8875
0.76	7	0.000285	2.55E-07	29.58371	1348.276	332.5744
0.77	5	0.000286	2.58E-07	23.52464	1354.163	213.0624
0.78	24.7	0.000288	2.6E-07	69.82787	1360	1901.613
0.78	6	0.000288	2.6E-07	26.6351	1360	276.6771
0.79	20	0.000289	2.62E-07	60.47743	1365.788	1444.72
0.79	1	0.000289	2.62E-07	7.860427	1365.788	24.4056
0.8	10	0.00029	2.64E-07	37.71883	1371.526	569.0839
0.8	6	0.00029	2.64E-07	26.6351	1371.526	283.7714
0.81	5	0.000291	2.66E-07	23.52464	1377.217	224.1306
0.81	1	0.000291	2.66E-07	7.860427	1377.217	25.02346
0.82	15	0.000292	2.69E-07	49.71598	1382.862	1013.388
0.82	4	0.000292	2.69E-07	20.20768	1382.862	167.4236
0.82	11	0.000292	2.69E-07	40.24864	1382.862	664.1808
0.85	6	0.000296	2.75E-07	26.6351	1399.524	301.5071
0.86	21	0.000297	2.77E-07	62.52095	1404.991	1680.814
0.88	6	0.000299	2.81E-07	26.6351	1415.799	312.1485
0.9	5	0.000302	2.86E-07	23.52464	1426.445	249.0339
0.91	2	0.000303	2.88E-07	12.60321	1431.708	72.27265
0.93	3	0.000305	2.92E-07	16.61189	1442.121	128.3191
0.94	2	0.000306	2.94E-07	12.60321	1447.272	74.65527
0.94	6	0.000306	2.94E-07	26.6351	1447.272	333.4314
0.97	5	0.000309	3E-07	23.52464	1462.507	268.4033
0.98	7.5	0.00031	3.02E-07	31.00708	1467.516	471.1052
0.98	7	0.00031	3.02E-07	29.58371	1467.516	428.8459
0.99	5	0.000311	3.04E-07	23.52464	1472.491	273.9373
0.99	7	0.000311	3.04E-07	29.58371	1472.491	433.2219
1	13	0.000312	3.07E-07	45.09894	1477.432	1016.957
1	46.4	0.000312	3.07E-07	107.2832	1477.432	5754.838
1.01	4	0.000313	3.09E-07	20.20768	1482.34	206.2169
1.01	4	0.000313	3.09E-07	20.20768	1482.34	206.2169
1.02	14	0.000314	3.11E-07	47.43378	1487.216	1147.481
1.02	6	0.000314	3.11E-07	26.6351	1487.216	361.8085
1.03	1	0.000315	3.13E-07	7.860427	1492.061	31.81995
1.04	26.7	0.000316	3.15E-07	73.6309	1496.874	2819.185
1.04	11	0.000316	3.15E-07	40.24864	1496.874	842.3756
1.06	9	0.000318	3.19E-07	35.10688	1506.409	653.2212
1.08	1	0.00032	3.23E-07	7.860427	1515.824	33.36461
1.08	2	0.00032	3.23E-07	12.60321	1515.824	85.77414
1.11	9	0.000323	3.29E-07	35.10688	1529.731	684.0335
1.11	18	0.000323	3.29E-07	56.28949	1529.731	1758.521
1.12	33.7	0.000324	3.31E-07	86.28459	1534.311	4169.217
1.14	65.4	0.000326	3.35E-07	135.537	1543.39	10471.06
1.14	15	0.000326	3.35E-07	49.71598	1543.39	1408.857
1.16	12	0.000328	3.38E-07	42.70606	1552.364	1057.808
1.2	1	0.000332	3.46E-07	7.860427	1570.006	37.07179
1.22	6	0.000334	3.5E-07	26.6351	1578.68	432.7514
1.23	1	0.000335	3.52E-07	7.860427	1582.982	37.99859
1.26	19	0.000337	3.58E-07	58.40104	1595.748	2148.729
1.27	25	0.000338	3.59E-07	70.40441	1599.958	3147.556
1.3	16	0.000341	3.65E-07	51.95014	1612.459	1754.231

1.3	5	0.000341	3.65E-07	23.52464	1612.459	359.7157
1.35	6	0.000345	3.74E-07	26.6351	1632.872	478.8642
1.39	2	0.000349	3.82E-07	12.60321	1648.842	110.3945
1.4	1	0.000349	3.84E-07	7.860427	1652.787	43.25042
1.41	11	0.00035	3.85E-07	40.24864	1656.713	1142.067
1.44	19	0.000353	3.91E-07	58.40104	1668.38	2455.691
1.48	2	0.000356	3.98E-07	12.60321	1683.687	117.5423
1.49	9	0.000357	4E-07	35.10688	1687.471	918.2071
1.51	2	0.000358	4.03E-07	12.60321	1694.987	119.925
1.59	8	0.000365	4.18E-07	32.40049	1724.407	834.5845
1.61	5	0.000366	4.21E-07	23.52464	1731.607	445.4941
1.66	16	0.00037	4.3E-07	51.95014	1749.35	2240.018
1.66	1	0.00037	4.3E-07	7.860427	1749.35	51.28265
1.72	1	0.000374	4.4E-07	7.860427	1770.178	53.13623
1.74	1	0.000376	4.43E-07	7.860427	1777.013	53.7541
1.75	16.5	0.000376	4.45E-07	53.05045	1780.411	2462.557
1.76	6	0.000377	4.47E-07	26.6351	1783.795	624.1971
1.77	2	0.000378	4.49E-07	23.52464	1787.167	489.7668
1.78	1	0.000379	4.5E-07	7.860427	1790.527	54.28982
1.83	16	0.000382	4.59E-07	51.95014	1807.137	2469.418
1.85	3	0.000383	4.62E-07	16.61189	1813.697	255.2584
1.86	12	0.000384	4.64E-07	42.70606	1816.959	1696.141
1.87	1	0.000385	4.65E-07	7.860427	1820.209	57.77021
1.88	2	0.000386	4.67E-07	12.60321	1823.448	149.3105
1.92	8	0.000388	4.74E-07	32.40049	1836.29	1007.8
1.93	15	0.000389	4.75E-07	49.71598	1839.472	2385.17
1.95	3	0.00039	4.78E-07	16.61189	1845.804	269.0562
1.98	37.7	0.000392	4.83E-07	93.13463	1855.222	8587.318
2.04	1	0.000396	4.93E-07	7.860427	1873.775	63.02205
2.14	9	0.000403	5.09E-07	35.10688	1903.906	1318.767
2.16	13	0.000404	5.12E-07	45.09894	1909.818	2196.628
2.16	2.5	0.000404	5.12E-07	14.67195	1909.818	232.4874
2.19	1	0.000406	5.17E-07	7.860427	1918.619	67.65602
2.21	6	0.000407	5.2E-07	26.6351	1924.442	783.9185
2.21	1	0.000407	5.2E-07	7.860427	1924.442	68.27388
2.2	5	0.000407	5.2E-07	23.52464	1924.442	611.5167
2.25	2	0.000409	5.26E-07	12.60321	1935.983	178.6961
2.26	19	0.00041	5.28E-07	58.40104	1938.847	3854.07
2.26	32	0.00041	5.28E-07	83.29556	1938.847	7840.111
2.3	2.5	0.000412	5.34E-07	14.67195	1950.219	247.5561
2.32	11	0.000414	5.37E-07	40.24864	1955.856	1379.146
2.34	2	0.000415	5.4E-07	12.60321	1961.46	185.844
2.35	9	0.000415	5.42E-07	35.10688	1964.25	1448.179
2.39	8	0.000417	5.46E-07	32.40049	1972.573	1249.252
2.38	11	0.000417	5.46E-07	40.24864	1972.573	1927.744
2.42	8	0.000419	5.53E-07	32.40049	1983.563	1270.248
2.48	16	0.000423	5.62E-07	51.95014	1999.822	3346.533
2.51	12	0.000425	5.66E-07	42.70606	2007.854	2288.878
2.55	11	0.000427	5.72E-07	40.24864	2018.463	2065.44
2.55	16	0.000427	5.72E-07	51.95014	2018.463	3440.992
2.56	23	0.000427	5.74E-07	66.51741	2021.098	5663.444
2.6	6	0.00043	5.8E-07	26.6351	2031.571	922.2571
2.61	17	0.00043	5.81E-07	54.14018	2034.172	3825.162
2.72	7	0.000436	5.97E-07	29.58371	2062.357	1190.266
2.77	27	0.000439	6.05E-07	74.19339	2074.917	7623.952
2.77	1	0.000439	6.05E-07	7.860427	2074.917	85.57405

2.79	1	0.00044	6.07E-07
2.82	4	0.000441	6.12E-07
2.82	1	0.000441	6.12E-07
2.82	9	0.000441	6.12E-07
2.84	1	0.000442	6.15E-07
2.84	13	0.000442	6.15E-07
2.87	8	0.000444	6.19E-07
2.87	1	0.000444	6.19E-07
3.04	5.5	0.000453	6.43E-07
3.08	6	0.000454	6.49E-07
3.1	5	0.000455	6.52E-07
3.13	11	0.000457	6.56E-07
3.48	1	0.000473	7.04E-07
3.53	1	0.000476	7.11E-07
3.57	4	0.000477	7.16E-07
3.77	5.5	0.000486	7.43E-07
3.79	73	0.000487	7.45E-07
4	2	0.000496	7.72E-07
4.08	1	0.000499	7.83E-07
4.08	2	0.000499	7.83E-07
4.15	1	0.000502	7.92E-07
4.3	9	0.000508	8.11E-07
4.36	9	0.00051	8.18E-07
4.44	7	0.000513	8.28E-07
4.59	30.5	0.000519	8.47E-07
4.76	12	0.000525	8.67E-07
4.92	12	0.000531	8.87E-07
4.94	1	0.000532	8.89E-07
5.07	3	0.000537	9.05E-07
5.09	20	0.000537	9.07E-07
5.11	2	0.000538	9.09E-07
5.54	29.5	0.000553	9.6E-07
5.59	14	0.000554	9.65E-07
6	1	0.000568	1.01E-06
6.6	1	0.000586	1.08E-06
6.78	25	0.000591	1.1E-06
7.28	49.4	0.000605	1.15E-06
7.78	10	0.000619	1.2E-06
7.85	8	0.000621	1.21E-06
8.1	17	0.000627	1.24E-06
8.35	28	0.000634	1.26E-06
8.99	1	0.00065	1.33E-06
9.66	9	0.000665	1.39E-06
10.99	8	0.000694	1.52E-06
11.05	8	0.000696	1.52E-06
11.75	33	0.00071	1.58E-06
12.6	49	0.000727	1.66E-06
14.29	15	0.000758	1.81E-06
16.07	11.5	0.000788	1.95E-06
17.36	1	0.000809	2.06E-06
17.6	1	0.000813	2.07E-06
25.03	6	0.000914	2.62E-06
25.1	24.7	0.000915	2.63E-06
25.21	28	0.000916	2.64E-06
25.76	25	0.000923	2.67E-06
26.48	1	0.000931	2.72E-06
28.19	34	0.000951	2.84E-06
28.55	20	0.000955	2.86E-06
29.33	67	0.000963	2.92E-06
29.84	50	0.000969	2.95E-06
45.26	55	0.001113	3.89E-06

7.860427	2079.899	86.19192
20.20768	2087.327	575.7739
7.860427	2087.327	87.11871
35.10688	2087.327	1737.815
12.60321	2092.25	243.8542
45.09894	2092.25	2888.159
32.40049	2099.591	1506.451
7.860427	2099.591	88.66337
25.10245	2140.254	957.802
26.6351	2149.6	1092.52
23.52464	2154.243	857.7836
40.24864	2161.17	2535.227
7.860427	2238.896	107.5082
7.860427	2249.568	109.0529
20.20768	2258.033	728.9053
25.10245	2299.436	1187.801
146.0756	2303.495	40435.67
12.60321	2345.277	317.682
7.860427	2360.809	126.0441
12.60321	2360.809	324.0356
7.860427	2374.234	128.2066
35.10688	2402.501	2649.859
35.10688	2413.624	2686.834
29.58371	2428.297	1942.934
80.61588	2455.34	14915.02
42.70606	2485.286	4340.661
42.70606	2512.827	4486.566
7.860427	2516.227	152.6122
16.61189	2538.108	699.5461
60.47743	2541.441	9308.386
12.60321	2544.766	405.8388
78.80605	2614.232	17202.79
47.43378	2622.073	6288.648
7.860427	2684.672	185.359
7.860427	2771.333	203.8949
70.40441	2796.302	16803.49
111.9603	2863.417	45627.77
37.71883	2927.525	5534.341
32.40049	2936.279	4120.433
54.14018	2967.125	11871.19
76.05414	2997.342	24149.17
7.860427	3072.044	277.7295
35.10688	3146.539	5952.94
32.40049	3284.783	5768.606
32.40049	3290.75	5800.1
85.05977	3358.82	42506.6
111.342	3437.935	78101.35
49.71598	3585.24	17660.15
41.48587	3728.316	13828.86
7.860427	3825.522	536.3053
7.860427	3843.071	543.7196
26.6351	4321.764	8878.498
69.82787	4325.789	61192.93
76.05414	4332.099	72910.25
70.40441	4363.377	63843.34
7.860427	4403.657	818.0509
86.80702	4496.478	106212.3
60.47743	4515.538	52211.09
137.7868	4556.291	278418
112.8847	4582.549	190124.8
120.4559	5265.169	328352.8

40.24	70	0.001101	3.95E-06	141.9595	5302.9	465926
48.4	72	0.001138	4.07E-06	144.7097	5384.217	506769.6
57.73	46	0.001207	4.58E-06	106.6524	5710.069	328331.5
73.77	9	0.00131	5.39E-06	35.10688	6196.323	45430.5
75.36	57	0.001319	5.47E-06	123.4223	6240.525	573981.8
77.17	1	0.00133	5.56E-06	7.860427	6290.092	2384.025
79.29	67	0.001342	5.66E-06	137.7868	6347.172	752668.4
101.88	68	0.001459	6.69E-06	139.1842	6900.339	986822.1
136.16	40	0.001607	8.11E-06	96.968	7600.765	640142.2
136.41	38	0.001608	8.12E-06	93.63878	7605.414	598036.5
186.3	64	0.001784	1E-05	133.554	8438.108	1661487
233.3	1	0.001923	1.16E-05	7.860427	9095.203	7207.374
268.55	64	0.002015	1.28E-05	133.554	9531.967	2395020

SHOT 6584

Rho = 31.04
Rhosubs = 7833
Beta =0.681112
Bsub1 =6.766796

Mass	Pene-tration	r	a	Calc Velocity	c	Kinetic Energy
0.09	8.3	0.00014	6.16E-08	33.22319	662.097	49.6701
0.09	4.8	0.00014	6.16E-08	22.87956	662.097	23.55635
0.1	36	0.000145	6.6E-08	90.25317	685.7631	407.2917
0.11	29	0.00015	7.04E-08	77.89382	707.8997	333.7096
0.11	6.3	0.00015	7.04E-08	27.5351	707.8997	41.69998
0.12	6	0.000154	7.46E-08	26.6351	728.7321	42.56571
0.13	2	0.000158	7.87E-08	12.60321	748.437	10.32466
0.14	4	0.000162	8.26E-08	20.20768	767.1556	28.58452
0.14	3	0.000162	8.26E-08	16.61189	767.1556	19.31685
0.14	29	0.000162	8.26E-08	77.89382	767.1556	424.7015
0.15	25	0.000166	8.65E-08	70.40441	785.0028	371.7586
0.17	4	0.000173	9.41E-08	20.20768	818.4468	34.70978
0.17	29.3	0.000173	9.41E-08	78.44175	818.4468	523.0143
0.17	54.9	0.000173	9.41E-08	120.3067	818.4468	1230.264
0.17	13.7	0.000173	9.41E-08	46.73908	818.4468	185.6861
0.18	7.7	0.000176	9.77E-08	31.5679	834.19	89.68788
0.19	2.6	0.00019	1.01E-07	15.06918	849.3604	21.5726
0.19	69.23	0.00018	1.01E-07	140.8941	849.3604	1885.858
0.2	14.9	0.000183	1.05E-07	49.48999	864.0074	244.926
0.21	24	0.000186	1.08E-07	68.47383	878.174	492.3098
0.22	9.8	0.000189	1.12E-07	37.20336	891.8977	152.2499
0.22	3	0.000189	1.12E-07	16.61189	891.8977	30.35506
0.23	1	0.000191	1.15E-07	7.860427	905.2115	7.105427
0.23	5	0.000191	1.15E-07	23.52464	905.2115	63.64201
0.23	37	0.000191	1.15E-07	91.95327	905.2115	972.3714
0.23	8.7	0.000191	1.15E-07	34.30552	905.2115	135.3399
0.24	16	0.000194	1.18E-07	51.95014	918.1449	323.8581
0.28	36	0.000204	1.31E-07	90.25317	966.5555	1140.389
0.28	17.6	0.000204	1.31E-07	55.43446	966.5555	430.217
0.28	32.1	0.000204	1.31E-07	83.47277	966.5555	975.4784
0.29	1	0.000207	1.34E-07	7.860427	977.9278	8.959016
0.3	37.5	0.000209	1.37E-07	92.79781	989.0416	1291.715
0.3	19.8	0.000209	1.37E-07	60.06485	989.0416	541.1679
0.32	14	0.000214	1.43E-07	47.43378	1010.549	359.9941
0.32	20.7	0.000214	1.43E-07	61.91122	1010.549	613.2799
0.32	39	0.000214	1.43E-07	95.3102	1010.549	1453.445
0.32	75.14093	0.000214	1.43E-07	148.9801	1010.549	3551.211
0.33	25	0.000216	1.46E-07	70.40441	1020.968	817.8689
0.34	67	0.000218	1.49E-07	137.7868	1031.178	3227.485
0.34	14	0.000218	1.49E-07	47.43378	1031.178	382.4938
0.35	3	0.00022	1.52E-07	16.61189	1041.19	48.29213
0.36	8.5	0.000222	1.55E-07	33.76638	1051.014	205.2304
0.36	10	0.000222	1.55E-07	37.71883	1051.014	256.0878
0.37	17	0.000224	1.58E-07	54.14018	1060.656	542.2644
0.39	27.3	0.000228	1.64E-07	74.75389	1079.433	1089.688
0.4	21	0.00023	1.66E-07	62.52095	1088.581	781.7739
0.4	1.3	0.00023	1.66E-07	9.398409	1088.581	17.66602
0.42	20.7	0.000234	1.72E-07	61.91122	1106.43	804.9298
0.43	54.2	0.000236	1.75E-07	119.2597	1115.142	3057.921
0.46	2.4	0.000241	1.83E-07	14.26963	1140.495	46.83311
0.47	38.8	0.000243	1.85E-07	94.97702	1148.7	2119.849

0.49	22	0.000246	1.91E-07	64.53367	1164.768	1020.326
0.49	21	0.000246	1.91E-07	62.52095	1164.768	957.6731
0.5	64.4	0.000248	1.93E-07	134.122	1172.638	4497.178
0.5	22.5	0.000248	1.93E-07	65.52905	1172.638	1073.514
0.5	19.8	0.000248	1.93E-07	60.06485	1172.638	901.9465
0.51	3	0.00025	1.96E-07	16.61189	1180.405	70.36854
0.51	9.7	0.00025	1.96E-07	36.94437	1180.405	348.046
0.52	17	0.000251	1.98E-07	54.14018	1188.07	762.1013
0.55	5	0.000256	2.06E-07	23.52464	1210.491	152.1874
0.57	3	0.000259	2.11E-07	16.61189	1224.99	78.64719
0.58	5	0.00026	2.13E-07	23.52464	1232.112	160.4885
0.59	36	0.000262	2.16E-07	90.25317	1239.153	2402.962
0.6	26	0.000263	2.18E-07	72.31052	1246.114	1568.643
0.63	33.54	0.000268	2.25E-07	86.00535	1266.546	2330.03
0.64	2	0.000269	2.28E-07	12.60321	1273.212	50.82912
0.65	4.7	0.000271	2.3E-07	22.55382	1279.809	165.3193
0.65	35.25	0.000271	2.3E-07	88.9682	1279.809	2572.486
0.66	11	0.000272	2.32E-07	40.24864	1286.339	534.5845
0.68	42.72002	0.000275	2.37E-07	101.4119	1299.203	3496.685
0.69	32.56	0.000276	2.39E-07	84.28565	1305.541	2450.905
0.71	7.8	0.000279	2.44E-07	31.84656	1318.035	360.0422
0.72	39	0.00028	2.46E-07	95.3102	1324.194	3270.252
0.72	7	0.00028	2.46E-07	29.58371	1324.194	315.0704
0.73	7.3	0.000281	2.49E-07	30.44148	1330.296	338.2395
0.74	35	0.000283	2.51E-07	88.53794	1336.343	2900.418
0.75	31	0.000284	2.53E-07	81.51368	1342.336	2491.68
0.78	1	0.000288	2.6E-07	7.860427	1360	24.09666
0.78	18	0.000288	2.6E-07	56.28949	1360	1235.718
0.79	35.25975	0.000289	2.62E-07	88.98496	1365.788	3127.738
0.8	26	0.00029	2.64E-07	72.31052	1371.526	2091.525
0.8	40	0.00029	2.64E-07	96.968	1371.526	3761.117
0.81	54.4	0.000291	2.66E-07	119.5593	1377.217	5789.244
0.81	28.6	0.000291	2.66E-07	77.16041	1377.217	2411.26
0.85	1	0.000296	2.75E-07	7.860427	1399.524	26.25919
0.85	30	0.000296	2.75E-07	79.71337	1399.524	2700.544
0.87	15	0.000298	2.79E-07	49.71598	1410.416	1075.18
0.92	12.9	0.000304	2.9E-07	44.86236	1436.934	925.8105
0.92	3.2	0.000304	2.9E-07	17.35841	1436.934	138.6048
0.94	18	0.000306	2.94E-07	56.28949	1447.272	1489.198
0.97	43.66	0.000309	3E-07	102.9264	1462.507	5138.016
0.98	54.6	0.00031	3.02E-07	119.8585	1467.516	7039.372
1.02	25	0.000314	3.11E-07	70.40441	1487.216	2527.958
1.02	44	0.000314	3.11E-07	103.4717	1487.216	5460.258
1.04	43	0.000316	3.15E-07	101.8641	1496.874	5395.673
1.04	31	0.000316	3.15E-07	81.51368	1496.874	3455.13
1.06	18	0.000318	3.19E-07	56.28949	1506.409	1679.309
1.09	5	0.000321	3.25E-07	23.52464	1520.488	301.6078
1.09	20.3	0.000321	3.25E-07	61.09384	1520.488	2034.189
1.11	42	0.000323	3.29E-07	100.2445	1529.731	5577.177
1.14	26	0.000326	3.35E-07	72.31052	1543.39	2980.423
1.14	8	0.000326	3.35E-07	32.40049	1543.39	598.3813
1.15	26	0.000327	3.36E-07	72.31052	1547.89	3006.567
1.15	3	0.000327	3.36E-07	16.61189	1547.89	158.6742
1.15	5	0.000327	3.36E-07	23.52464	1547.89	318.21
1.23	25	0.000335	3.52E-07	70.40441	1582.982	3048.42
1.25	24	0.000336	3.56E-07	68.47383	1591.515	2930.416

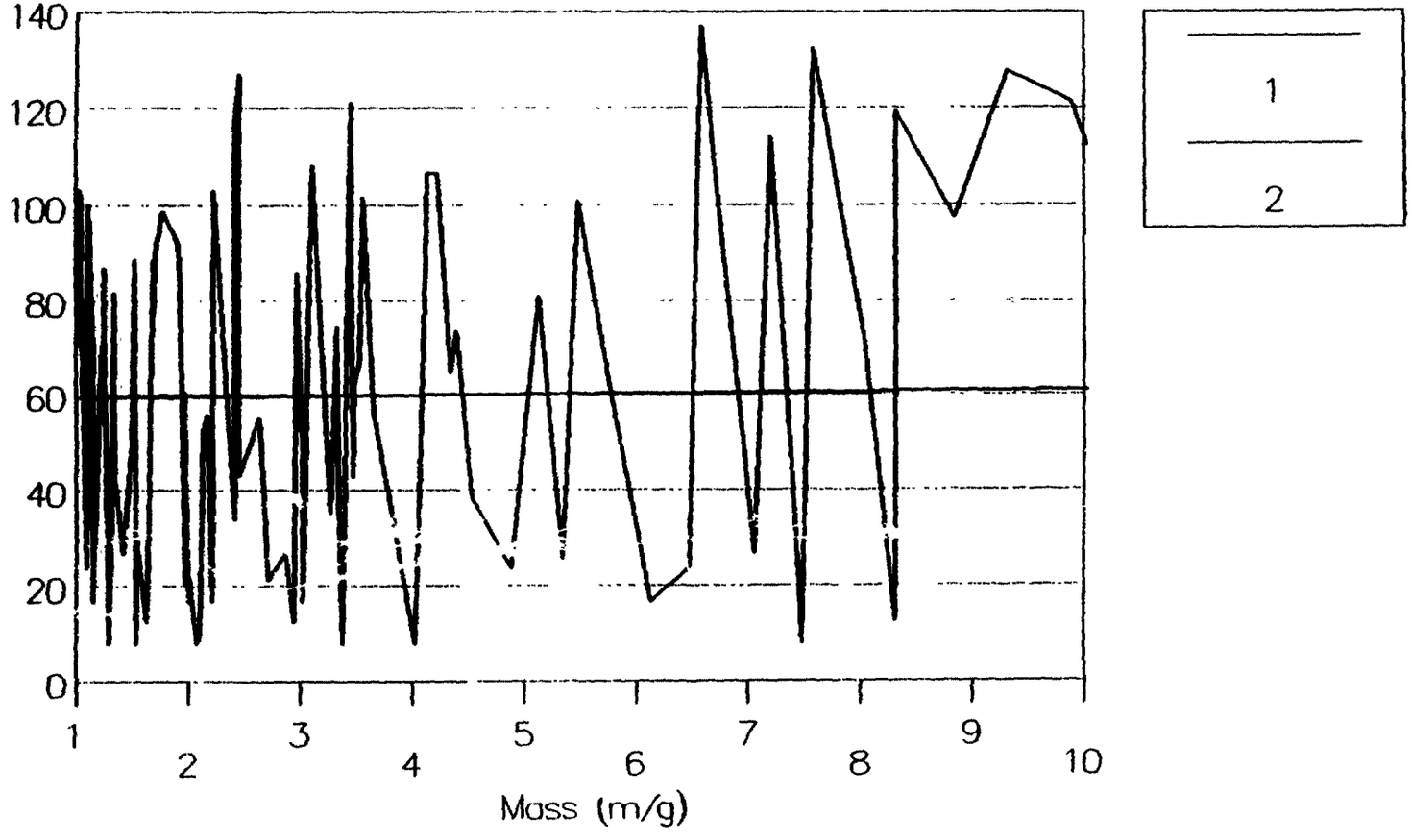
1.25	34	0.000336	3.56E-07	86.80702	1591.515	4709.661
1.25	16.7	0.000336	3.56E-07	53.48759	1591.515	1788.076
1.26	20	0.000337	3.58E-07	60.47743	1595.748	2304.237
1.29	1	0.00034	3.63E-07	7.860427	1608.313	39.85218
1.34	31	0.000344	3.73E-07	81.51368	1628.83	4451.802
1.34	12	0.000344	3.73E-07	42.70606	1628.83	1221.951
1.42	6	0.000351	3.87E-07	26.6351	1660.62	503.6942
1.5	15	0.000358	4.02E-07	49.71598	1691.237	1853.759
1.52	26.5	0.000359	4.05E-07	73.25479	1698.721	4078.36
1.53	35	0.00036	4.07E-07	88.53794	1702.438	5996.81
1.53	3	0.00036	4.07E-07	16.61189	1702.438	211.1056
1.54	1	0.000361	4.09E-07	7.860427	1706.139	47.57547
1.54	7	0.000361	4.09E-07	29.58371	1706.139	673.9007
1.63	2	0.000368	4.25E-07	12.60321	1738.748	129.4554
1.7	35	0.000373	4.37E-07	88.53794	1763.29	6663.123
1.7	28.5	0.000373	4.37E-07	76.97655	1763.29	5036.581
1.72	36.3	0.000374	4.4E-07	90.76476	1770.178	7084.888
1.79	41	0.000379	4.52E-07	98.61264	1793.873	8703.386
1.91	37.3	0.000388	4.72E-07	92.46043	1833.096	8164.229
1.93	35	0.000389	4.75E-07	88.53794	1839.472	7564.604
1.98	4	0.000392	4.83E-07	20.20768	1855.222	404.2668
1.98	20	0.000392	4.83E-07	60.47743	1855.222	3620.944
1.99	7	0.000393	4.85E-07	29.58371	1858.34	870.8197
2.01	3	0.000394	4.88E-07	16.61189	1864.545	277.3348
2.01	4.5	0.000394	4.88E-07	21.89561	1864.545	481.8148
2.08	1	0.000399	4.99E-07	7.860427	1885.943	64.25777
2.11	1.5	0.000401	5.04E-07	10.36058	1894.967	113.2454
2.14	16	0.000403	5.09E-07	51.95014	1903.906	2887.734
2.18	17.8	0.000405	5.15E-07	55.86274	1915.695	3401.504
2.22	3	0.000407	5.22E-07	16.61189	1927.341	306.3101
2.24	43.68	0.000409	5.25E-07	102.9585	1933.111	11872.51
2.42	8.5	0.000419	5.53E-07	33.76638	1983.563	1379.604
2.42	52.6	0.000419	5.53E-07	116.8504	1983.563	16521.36
2.46	59.5	0.000422	5.59E-07	127.084	1994.432	19864.93
2.46	12.08305	0.000422	5.59E-07	42.90713	1994.432	2264.457
2.65	17.49286	0.000432	5.87E-07	55.20438	2044.511	4037.968
2.72	4.3	0.000436	5.97E-07	21.228	2062.357	612.8542
2.88	6	0.000444	6.2E-07	26.6351	2102.027	1021.577
2.95	2	0.000448	6.31E-07	12.60321	2118.921	234.2905
2.98	33.5	0.00045	6.35E-07	85.93547	2126.08	11003.51
3.03	14.9	0.000452	6.42E-07	49.48999	2137.905	3710.628
3.03	3	0.000452	6.42E-07	16.61189	2137.905	418.0719
3.12	47	0.000456	6.54E-07	108.2261	2158.866	18272.11
3.28	9	0.000464	6.77E-07	35.10688	2195.156	2021.288
3.34	27	0.000467	6.85E-07	74.19339	2206.46	9192.78
3.35	10	0.000467	6.86E-07	37.71883	2210.662	2383.039
3.38	1	0.000469	6.9E-07	7.860427	2217.242	104.4189
3.39	2	0.000469	6.92E-07	12.60321	2219.426	269.2355
3.46	55.4	0.000472	7.01E-07	121.0519	2234.599	25350.66
3.48	12	0.000473	7.04E-07	42.70606	2238.896	3173.424
3.49	21	0.000474	7.05E-07	62.52095	2241.038	6820.977
3.54	23	0.000476	7.12E-07	66.51741	2251.69	7831.482
3.57	42.79311	0.000477	7.16E-07	101.53	2258.033	18400.39
3.66	18	0.000481	7.28E-07	56.28949	2276.85	5798.368
4.03	1	0.000497	7.76E-07	7.860427	2351.126	124.4994
4.12	30	0.000501	7.88E-07	79.71337	2368.499	13089.7

4.14	46	0.000502	7.9E-07	106.6524	2372.325	23545.68
4.23	46	0.000505	8.02E-07	106.6524	2389.393	24057.55
4.35	22	0.00051	8.17E-07	64.53367	2411.777	9057.993
4.4	26.5	0.000512	8.23E-07	73.25479	2420.983	11805.78
4.54	10.2	0.000517	8.4E-07	38.23102	2446.392	3317.856
4.89	5	0.00053	8.83E-07	23.52464	2507.709	1353.084
5.14	30.6	0.000539	9.13E-07	80.79581	2549.736	16776.87
5.35	5.6	0.000546	9.38E-07	25.41242	2583.997	1727.491
5.49	42.2019	0.000551	9.54E-07	100.5725	2606.343	27765.2
5.78	20	0.000561	9.87E-07	60.47743	2651.45	10570.23
6.13	3	0.000572	1.03E-06	16.61189	2703.923	845.8022
6.48	5	0.000582	1.07E-06	23.52464	2754.435	1793.044
6.58	66.4	0.000585	1.08E-06	136.9452	2768.531	61700.59
7.07	6	0.0006	1.13E-06	26.6351	2835.615	2507.83
7.21	50.7	0.000603	1.14E-06	113.9587	2854.21	46816.65
7.49	1	0.000611	1.17E-06	7.860427	2890.689	231.3898
7.59	63.01143	0.000614	1.18E-06	132.1455	2903.497	66269.9
8.04	26	0.000626	1.23E-06	72.31052	2959.78	21019.82
8.32	2	0.000633	1.26E-06	12.60321	2993.748	660.7786
8.33	54.2	0.000633	1.26E-06	119.2597	2994.947	59238.33
8.85	40.2	0.000646	1.31E-06	97.29797	3056.013	41891.01
9.32	59.9363	0.000657	1.36E-06	127.718	3109.182	76013.39
9.89	55.6	0.00067	1.41E-06	121.3494	3171.316	72818.44
10.98	15	0.000694	1.51E-06	49.71598	3283.787	13569.52
11.15	25	0.000698	1.53E-06	70.40441	3300.647	27634.06
12.47	76.35208	0.000724	1.65E-06	150.6115	3426.07	141433.6
12.74	3	0.00073	1.67E-06	16.61189	3450.621	1757.834
12.92	42	0.000733	1.69E-06	100.2445	3466.796	64916.33
13.68	27	0.000747	1.75E-06	74.19339	3533.482	37651.87
14.63	15	0.000764	1.83E-06	49.71598	3613.452	18080.33
14.99	22	0.00077	1.86E-06	64.53367	3642.85	31213.64
15.29	1.5	0.000775	1.89E-06	10.36058	3666.992	820.6265
16.6	4	0.000797	1.99E-06	20.20768	3768.861	3389.308
16.93	52.6	0.000802	2.02E-06	116.8504	3793.672	115581.2
17.91	13.8	0.000817	2.1E-06	46.97118	3865.503	19757.34
21.8	13	0.000873	2.39E-06	45.09894	4127.239	22169.67
22.52	3	0.000882	2.44E-06	16.61189	4172.185	3107.254
30.56	17	0.000977	3E-06	54.14019	4619.113	44788.11
37.59	33.7	0.001046	3.44E-06	87.33968	4944.161	86626.32
38.41	50.5	0.001054	3.49E-06	113.6523	4984.39	248068.2
44.94	68	0.001111	3.87E-06	139.1342	5252.731	435294.3
59.77	27	0.001221	4.69E-06	74.19339	5776.55	164506.7
111.78	53.8	0.001505	7.11E-06	118.6596	7116.976	786936.3
193.21	13	0.001806	1.02E-05	45.09894	8541.169	196486.3

APPENDIX C

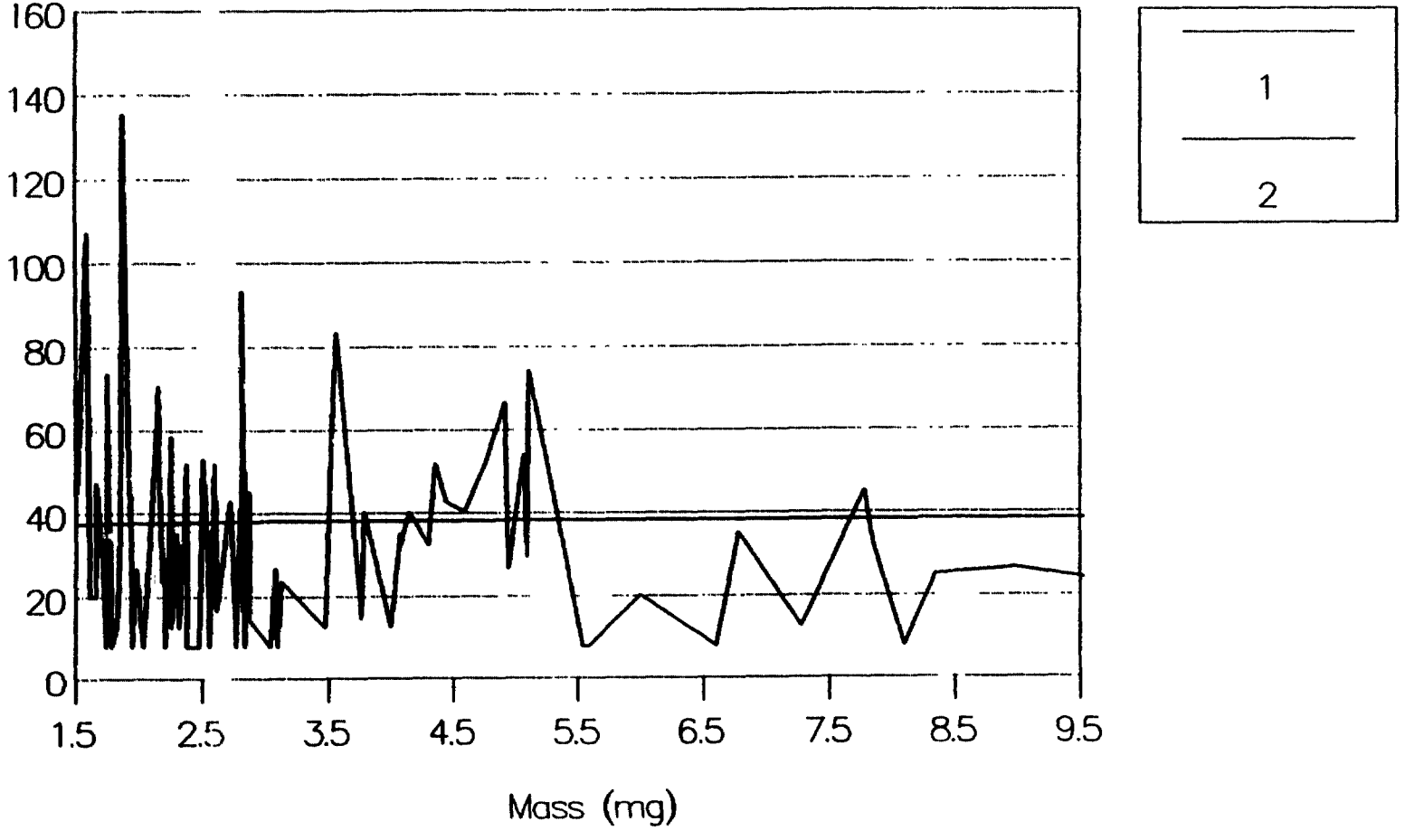
Velocity Distribution

Shot 6584



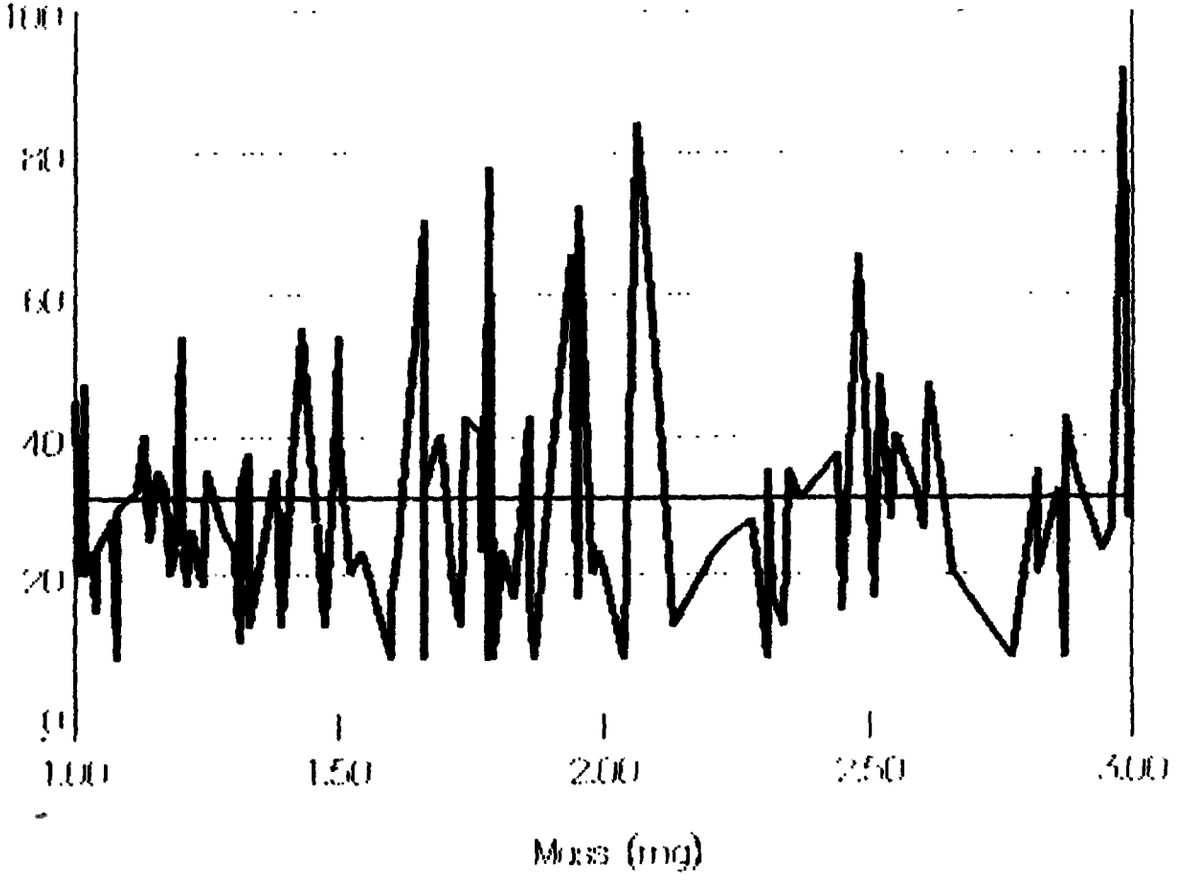
Velocity Distribution

Shot 6579



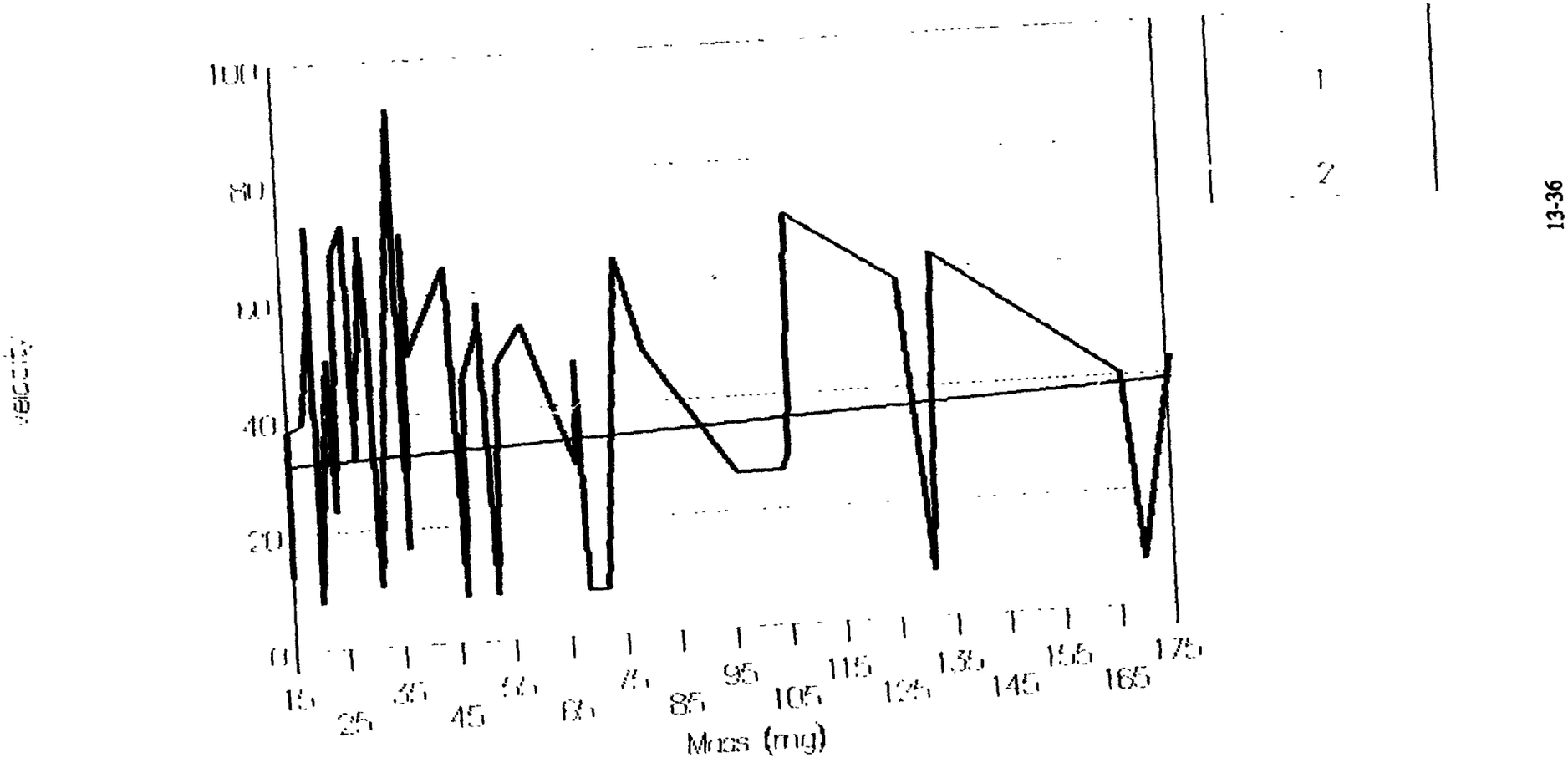
Velocity Distribution

Shot 6575



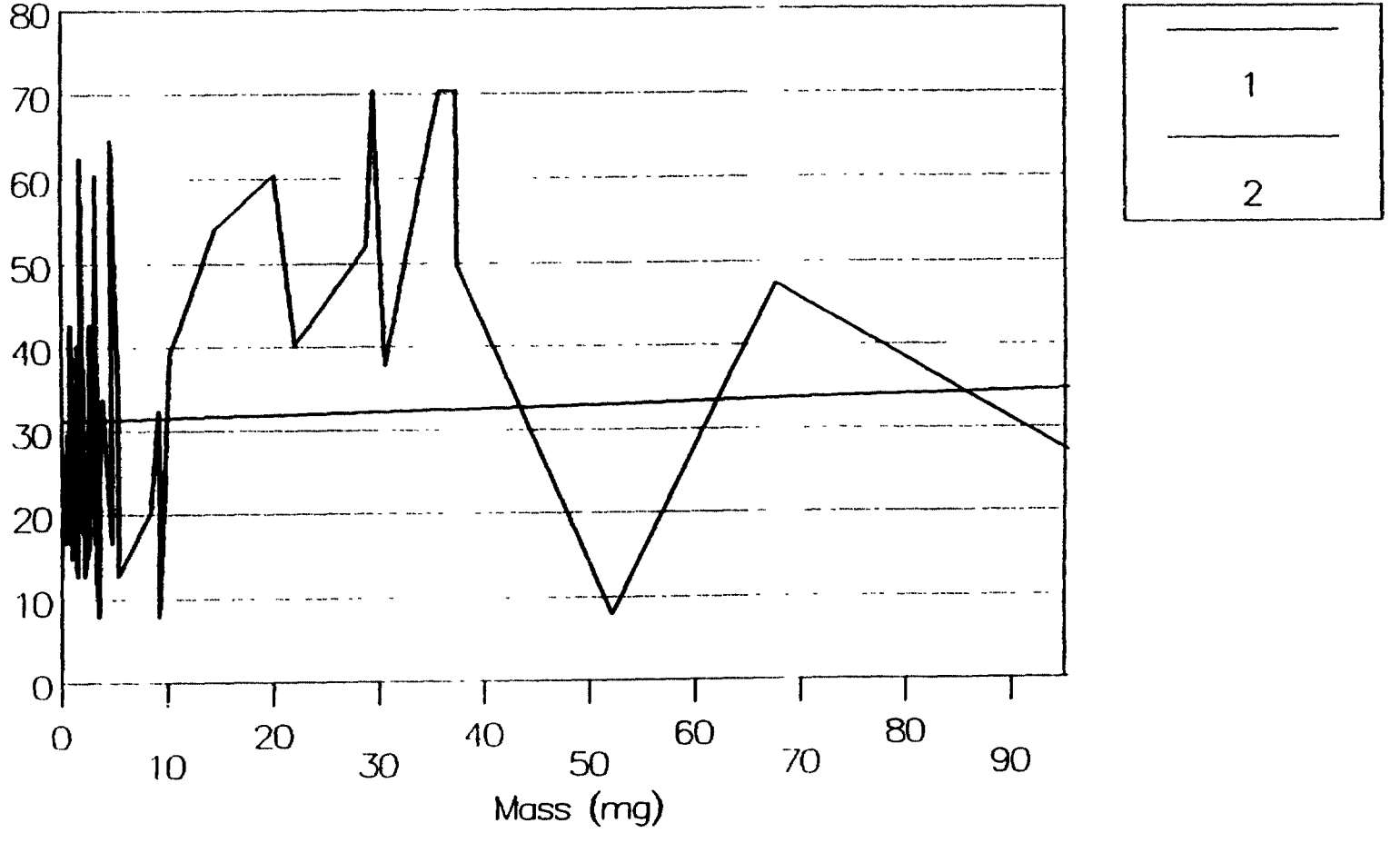
Velocity Distribution

Shot 6575



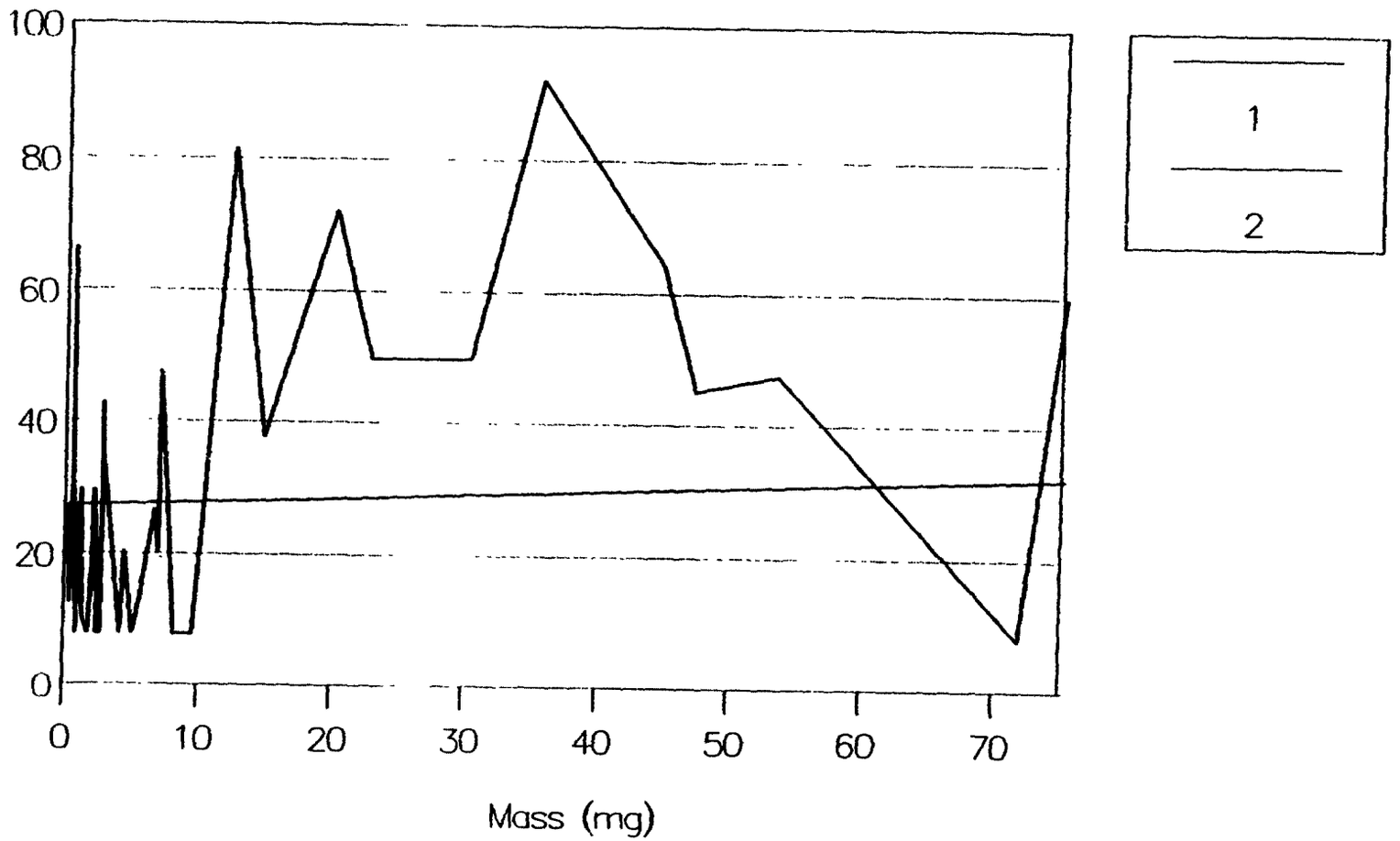
Velocity Distribution

SHOT 6575 SEAT 1B



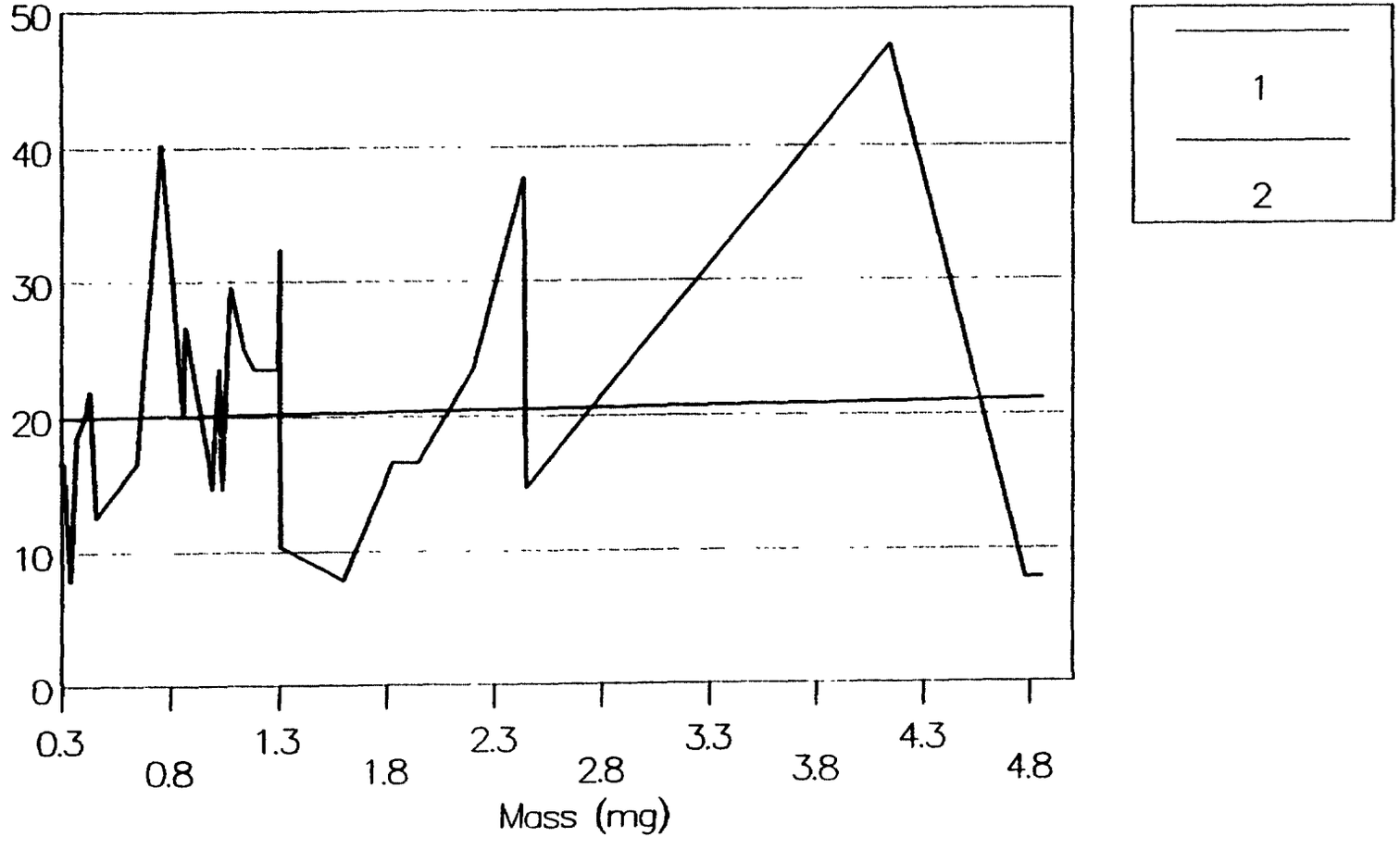
Velocity Distribution

SHOT 6575 SEAT 1D



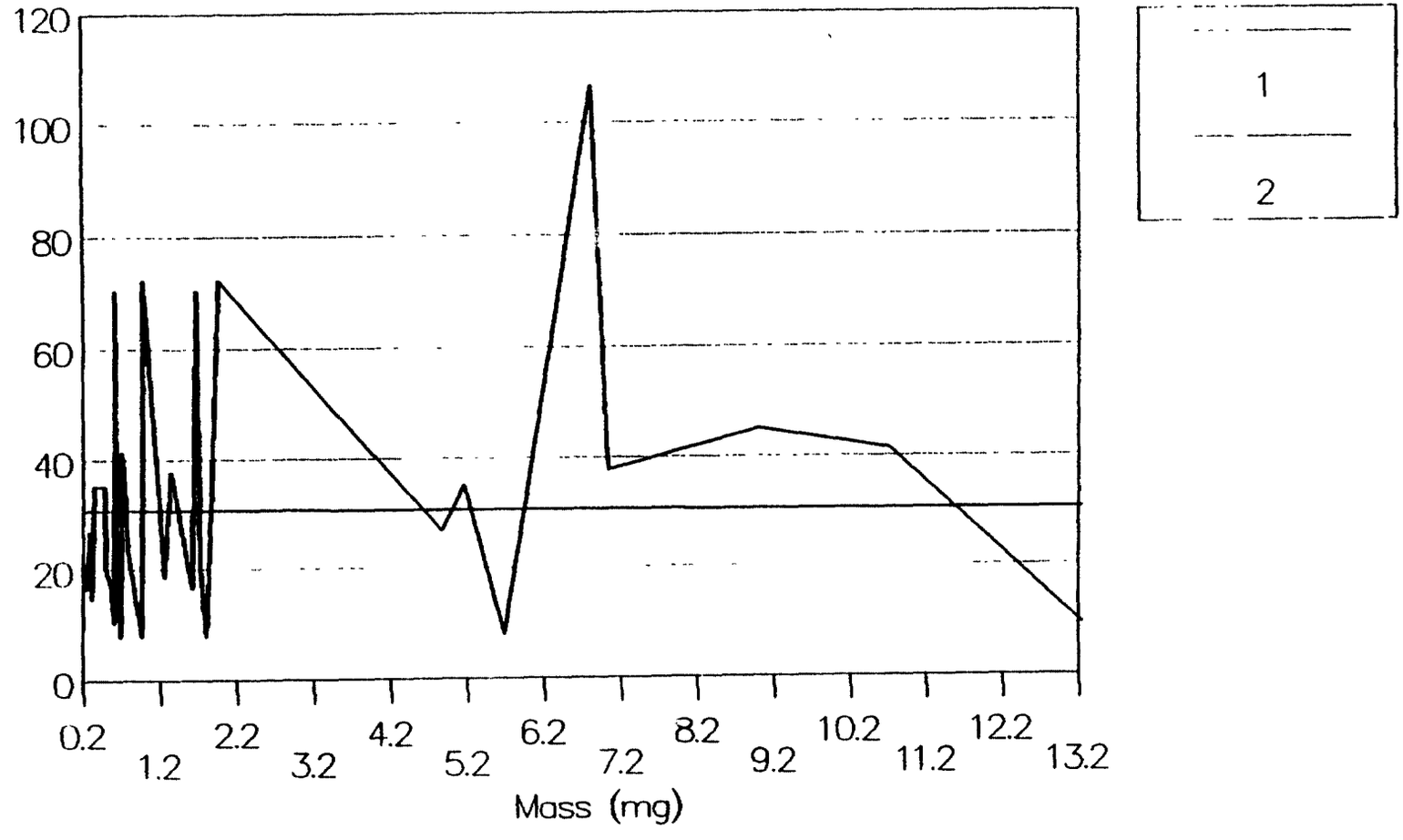
Velocity Distribution

SHOT 6575 SEAT 2B



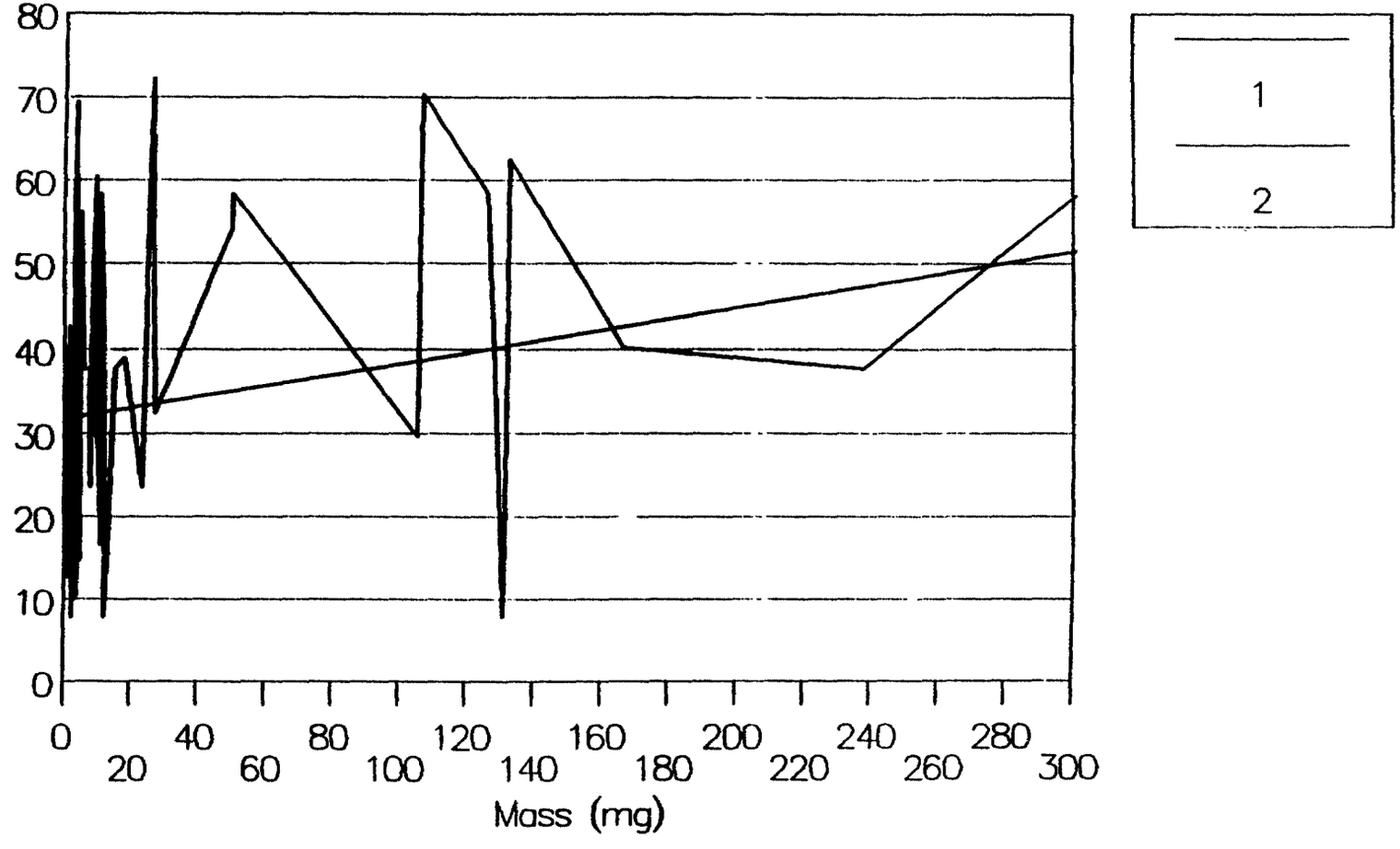
Velocity Distribution

SHOT 6575 SEAT 2D



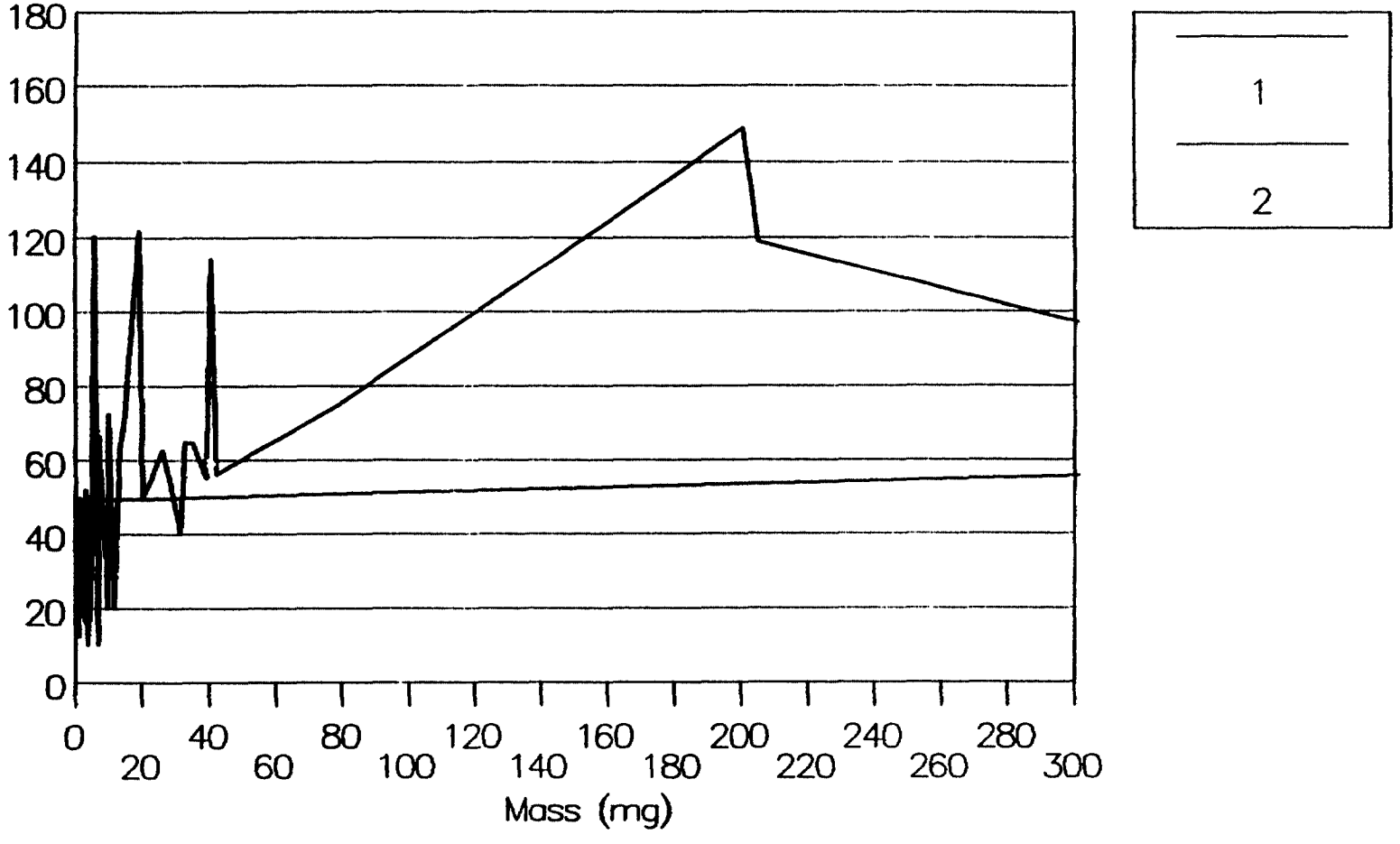
Velocity Distribution

SHOT 6575 SEAT 3B



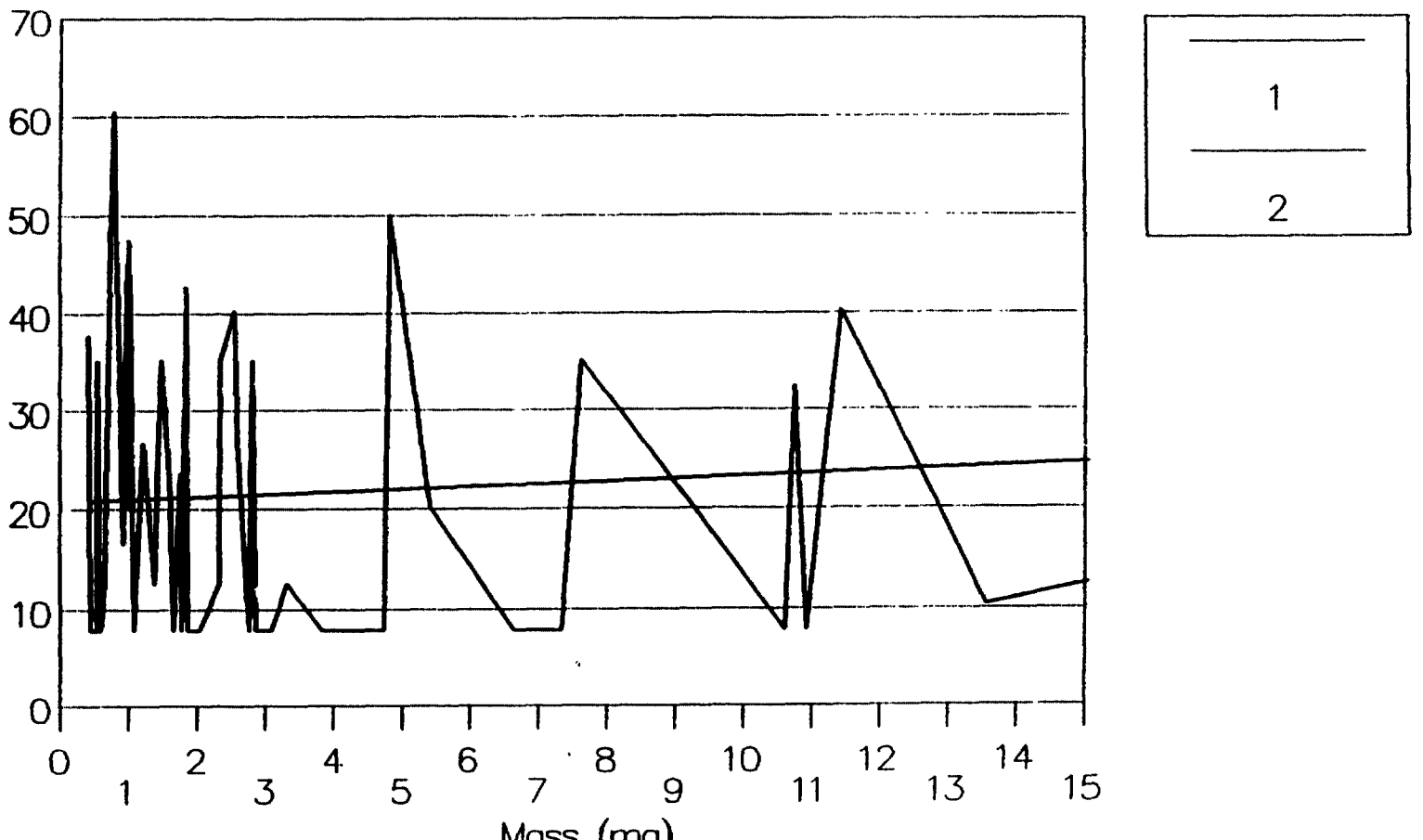
Velocity Distribution

SHOT 6575 SEAT 3D



Shot 6575

SEATs 4B, 4D, and 5B



ENHANCEMENT AND INTEGRATION OF POST PROCESS UTILITIES FOR THE EPIC HYDROCODES

KENNETH GAGE
HIGH SCHOOL APPRENTICESHIP PROGRAM
AFATL/MNW
MENTOR: MICHAEL E NIXON
SUMMER 1990

SECTION 1 - INTRODUCTION

This summer I participated in the High School Apprenticeship Program for the second time. As luck would have it, I was given the chance to work under Mike Nixon, who served as my mentor last year when Cpt. Mark Hand was absent. Under his direction I broadened my horizons and knowledge in the scientific and technical fields. In addition, I now understand more fully the work done at AFATL in general and MNW in particular.

Due to Mike Nixon's upcoming departure to the University of Florida to pursue his master's degree in engineering mechanics, my project this summer focused on creating a user-friendly environment for his temporary replacements to work in during his abscence. This required the modification of numerous post-process EPIC utilities and their subsequent integration into an easy access information and initiation tree called PHENIX. To complete my tasks, I had to develop an understanding of various computer programming languages, graphical environments, and the EPIC hydrocodes.

SECTION 2 - EPIC HYDROCODES

The EPIC hydrocodes are extremely large FORTRAN programs that simulate the interaction of matter and energy in nature. There are two EPIC hydrocodes: EPIC2 and EPIC3. EPIC2 is a two dimensional hydrocode and EPIC3 simulates interactions three dimensionally. Both hydrocodes have

their own separate utilities for processing inputs and outputs, but generally they perform the same function, just on a different scale. EPIC3 output files are enormous and dwarf the EPIC2 outputs in both size and complexity. This leads to problems that will be discussed later under the section about TRANIX, the EPIC3 file size reducing program.

EPIC calculations are characteristically run through three stages: a pre-processor, the main program, and the post-processor. The pre-processor is basically the input data deck. The input data deck is a file that contains values for all the variables and coordinates necessary to simulate an interaction. The pre-processor establishes the initial geometry for a model, including size and shape. It also includes an in-depth description of the materials used in the model, specifically the physical and chemical characteristics. The initial conditions, such as velocity, acceleration, and temperature, are also found in the pre-processor.

Most of the actual processing of EPIC is done in the main program. It is here where all the interactions are actually simulated. The main program of EPIC takes the original model under the initial conditions and runs it through a series of calculations that are aimed at simulating the real world. These calculations are based on the fundamental conservative equations of mass, momentum, energy, and entropy, along with the specific traits unique to that material, such as its failure criterion and

stress-strain handling characteristics. The use of precise, proven equations along with the actual, real-world data of the material allows for very accurate simulations to be produced.

The post-processor section of the EPIC hydrocode is what gives the user all the data from the simulation dealing with deformation, stress, temperature, pressure, and a dozen other variables This allows the user to actually view what is happening in the simulation as it takes place, for example, what pressure waves are being formed, what forces are acting upon the body, and what magnitude of stress the model is undergoing.

SECTION 3 - GRAPHICS PACKAGES

To actually view the output of an EPIC simulation, an EPIC compatible graphics package is needed. In MNW, two graphics packages are used: PATRAN and GKS. PATRAN is an extremely large, multipurpose graphical environment developed by PDA Engineering Associates to explore the outer limits of interactive engineering. PATRAN allows the user to locate very specific areas on an image with ease and modify those areas in nearly any way, shape, or form. The resolution of PATRAN is superb. In conjunction with a good quality wax printer, PATRAN can produce pictures that rival true photographs. PATRAN's few disabilities include its complexity and its uncompatibility with EPIC when unmodified. EPIC outputs must first be run through an EPIC

to PATRAN translator to be changed into a format that is readable by PATRAN.

The other graphics package used by MNW is GKS, or the Graphical Kernal System. GKS is considered the world standard in computer graphics and consists of hundreds of FORTRAN programs in a library which can be incorporated into other programs using FORTRAN call statements. GKS is not as precise or resolute as PATRAN, but it has one big advantage; it can be used to tailor-make a graphical program to the user's specifications. On the down side, the user must be both a FORTRAN and GKS expert to create a truly effective and useful program.

SECTION 4 - EPIC UTILITIES

My project this summer required the modification, documentation, and integration of most of the EPIC utilities. These numerous utilities performed many widely varied functions, such as data translation, results plotting, and file size reduction. I had to learn the inner workings of both the utilities and hydrocodes if I was to modify, document, and integrate them. This required that I expand my programming skills and actually use the hydrocodes and utilities to get the "feel" for what was going on inside the program. The utilities I dealt with over the course of the summer included TRAN2, EP3PAT, PATEP2, PATEP3, XYPLOT, and TRANIX. TRAN2 and EP3PAT are EPIC to PATRAN translators and were by far the most used utilities belonging to EPIC.

PATEP2 and PATEP3 are PATRAN to EPIC translators that create an EPIC input data deck from a PATRAN model file. The last two utilities were developed by former apprentices of Mike Nixon during the last few years. XYPLOT is a GKS based EPIC2 results plotter developed by Bryan McGraw last year and TRANIX is a EPIC3 output file size reduction program developed by former apprentice Neil Overholtz. Both are useful programs that increase the overall number of simulations produced due to reduced "idle" time on the keyboard.

SECTION 4.1 - EPIC TO PATRAN TRANSLATORS

The two EPIC to PATRAN translators, TRAN2 and EP3PAT, are identical in function. They simply change an EPIC output file into a form that is readable by PATRAN. EP3PAT required no modification to be incorporated into the PHENIX integrator, but TRAN2 was a different matter altogether. TRAN2 is a program developed from the original EPIC2 to PATRAN translator, EP2PAT. EP2PAT failed to work correctly with the latest versions of EPIC and therefore my mentor and I decided that it would be best to design a new translator. We began by modifying the main processing routines found in EP2PAT to handle the output of the latest versions of EPIC. Once this was completed, Mr. Nixon evaluated the effectiveness and necessity of each of the extraneous features of EP2PAT. As a result, many nice-to-have features that were never used were eliminated, and a compact, efficient translator emerged. Improvements over the

original design were also added. For example, the original translator would produce either a neutral file or a results file, not both. TRAN2 was designed to produce both files because a results file cannot be plotted unless PATRAN has a neutral file in its active set. This new, improved translator was quickly incorporated into my integrator and work moved on.

SECTION 4.2 - PATRAN TO EPIC TRANSLATORS

The next set of translators I dealt with were the PATRAN to EPIC translators. There are two of these, one for EPIC2 and one for EPIC3. They are nearly identical in performance, the only difference being that PATEP3 translates three dimensional plots and PATEP2 translates two dimesional ones. These translators can save a great deal of time by eliminating the need to rewrite an entire input deck once a modification has been made. A user may take an EPIC output, translate it, and plot it using PATRAN. He may now modify the plot, use the PATRAN to EPIC translator, and create an entirely new input data deck. The new data deck can now be processed by the EPIC hydrocodes and the new outputs compared with the originals. In this way, the effects of modifications on penetrators and targets can be evaluated quickly and easily.

SECTION 4.3 - XYPLOT

XYPLOT is a data plotting utility developed by Mike

Nixon's former apprentice, Bryan McGraw. It is designed to plot many of the variables used by the EPIC2 hydrocode, such as stress, pressure, or temperature, and its relation to the time-history of the model. This allows the user to view the changes that occur in the variables as the model moves from one time period to the next. Any possible errors that could be taking place in the simulation will also be revealed.

XYPLOT is based on GKS, or the Graphical Kernal System. It is an example of a tailor made FORTRAN program designed to meet the needs of the user by incorporating GKS subroutines. GKS is a fickle set of programs, however. It must be configured for exactly the equipment it is being run on or else it will terminate or produce erroneous results. At the beginning of the summer, XYPLOT was configured to run only on machines using the VMS operating system. With my mentor's help I modified the program to be run on UNIX based machines. Unfortunately, that was not the end of the program's faults. A few vitally important statements in XYPLOT were written in VAX FORTRAN, which will not compile on a UNIX based machine. I had to track the statements down and develop counterparts for them out of FORTRAN77. This proved more difficult than I would have thought but eventually the problems were overcome and XYPLOT was incorporated into PHENIX.

SECTION 4.4 - TRANIX

TRANIX is an EPIC3 file size reducing utility designed by another of Mike Nixon's former apprentices, Neil Overholtz. TRANIX is configured to remove all the interior nodes and elements from a three dimensional plot because the interior of the model will generally not be viewed anyway. TRANIX does this through a series of FORTRAN commands and system routines that first divides the tetrahedral elements into four faces. TRANIX then determines the interior faces and nodes and removes them. After all this has been done, TRANIX makes a new neutral file that includes only the exterior nodes and faces.

The reduction of file size and plot time is remarkable when using TRANIX. A three dimensional PATRAN neutral file of approximately nineteen thousand blocks of data is reduced to less than one third of its original size. Subsequently, the plot time is reduced from over three hours to seven minutes, forty-five seconds.

The modifications I performed upon were relatively simple and included removing extraneous features that identified slave nodes and assigned separate property ID's for the nodes and elements. These modifications slimmed the program up somewhat and gave me much-needed experience in deciding what was a necessary section of a program and what was in excess.

SECTION 5 - PHENIX INTEGRATOR

The main portion of my project this summer involved the design and development of a user-friendly program to integrate all major post process utilities for the EPIC hydrocodes. It was created primarily for use by the individuals who are to temporarily replace my mentor while he studies at the University of Florida. They lack the experience necessary to use the utilities found in PHENIX effectively without help and it is hoped that this program will allow them to overcome their barriers quickly.

PHENIX was designed around four requirements laid down by Mike Nixon at the beginning of the summer. They are as follows:

1. The program had to be accessible from anywhere on the system.

2. The program had to contain both a general overview for a utility and step-by-step instructions on how to use it.

3. PHENIX had to provide quick, easy access to all utilities so as to reduce any transfer time between directories.

4. Finally, the program had to be structured to allow for easy modification and incorporation of new utilities.

The first requirement was fairly easy to meet. I altered all the users' .login files by creating an 'alias'. An 'alias' is a UNIX command that takes an input, 'pi' in this case, and lets it stand for another, more complicated command. For example, once 'pi' was entered, the computer would automatically route itself to the PHENIX directory and initiate the TREE program. The .login file is simply a file that the computer reads when a user is first logging on. It sets the initial conditions for the operating environment, including paths for commands to follow and alias commands. It can be modified to fit the needs of the user.

The next requirement proved to be more difficult. PHENIX is written in a command shell format, which is simply a file filled with UNIX commands that the computer reads and initiates. The two UNIX commands able to print text are 'echo' and 'more'. 'Echo' will print only one line at a time , but after that line is printed the user may input data on what is to be done next, an important feature used often in interactive programs. If 'more' is used, a whole text file will be printed, but no interaction is allowed until after the entire file has been printed. 'Echo' proved difficult to work with because I had to type a command for

each line of text and 'echo' doesn't center automatically. Therefore I made sure that all questions would be asked after the entire text file had been printed. This allowed me to use the simpler, more efficient 'more' command for documentation purposes.

The last two requirements, utility access and simplified modification, were relatively easy to meet. To provide easy access I placed menus at the beginning of the program and after each documentation text which would route the computer to more documentation, the utility itself, or an exit. PHENIX's ease of modification is based on the fact that each utility has its own directory. To add a utility, you simply add a new directory, place the documentation in that directory, add a menu subprogram, and include a new choice in the top menu. Removing a utility entails removing that directory dealing with the utility and deleting that choice from the top menu.

SECTION 6 - RESULTS AND CONCLUSIONS

The development and implementation of the PHENIX Integrator led to four obvious conclusions. First, the use of PHENIX reduced the training time required for the TEAS replacements, thus allowing Mike Nixon to focus on the more important and subtle workings of the EPIC Hydrocodes. This reduction in training time led to an overall increase in the knowledge of the TEAS personnel about their duties, thus allowing a smoother transdition when Michael Nixon leaves

for school. PHENIX's flexibility and speed will provide greater time to be spent doing actual EPIC runs, thus allowing an increase in production of simulations. The last conclusion spawned by the development of the PHENIX Integrator is my own increase in knowledge. I had to learn the ins and outs of FORTRAN, GKS, and the EPIC utilities. This learning also led to a greater understanding of the functions that MNW performs in the overall mission of AFATL. I understand more fully the importance of the work that takes place at Eglin AFB and how its affects both our military and civilian organiztions not just here in the Panhandle, but around the world.

SECTION 7 - ACKNOWLEDGEMENTS

In the two years I have worked as an apprentice here at AFATL/MNW I learned more, experienced more, and had more fun than I ever had before. Much of this great experience hinged upon the attitude and abilities of our mentors. I would like to thank my mentor, Michael Nixon, for all his help and advice on everything ranging from the EPIC hydrocodes to colleges. I have enjoyed talking and working with the various engineers in my section, including Mr. Bill Cook, Ms. Lori Perillo, Mr. Bizhan Aref, and the TEAS personnel. I must also thank my fellow apprentices, especially Heather Cox, for all the good times and fun we shared. It is an memory I will hold throughout college and beyond. Again, I would like to thank Michael Nixon for everything he has helped me with. I greatly enjoy his

friendship and tutelage and I look forward to attending the University of Florida with him in the fall. The High School Apprenticeship Program is a great opportunity for students in this area and I would like to thank everyone who supports and organizes this program, especially Mr. Don Harrison and Dr. Norm Klausutis. Our thanks to them are well deserved.

BIBLIOGRAPHY

ECONET FORTRAN USERS GUIDE, MAY 1985

ULTRIX-32 QUICK REFERENCE, COPYRIGHT 1987, DIGITAL EQUIPMENT CORPORATION, MERRIMACK, NEW HAMPSHIRE.

ULTRIX-32 SUPPLEMENTARY DOCUMENTS, VOLUME 1, GENERAL USER, COPYRIGHT 1984, DIGITAL EQUIPMENT CORPORATION, MERRIMACK, NEW HAMPSHIRE.

VAX-11 FORTRAN LANGUAGE REFERENCE MANUAL, COPYRIGHT 1982, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.

VAX-11 FORTRAN USER'S GUIDE, COPYRIGHT 1982, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.

VAX FORTRAN FOR ULTRIX SYSTEMS USER MANUAL, COPYRIGHT 1986, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.

UNDERSTANDING FORTRAN: SECOND EDITION, MICHEL BOILLOT, MAY 1985.

INTRODUCTION TO FORTRAN, COPYRIGHT 1986, GOULD ELECTRONICS INC., FORT LAUDERDALE, FLORIDA.

INTRODUCING THE UNIX SYSTEM, COPYRIGHT 1983, HENRY MCGILTON AND RACHEL MORGAN, MCGRAW-HILL PUBLISHING, NEW YORK, NEW YORK.

UNIX UTILITIES: A PROGRAMMER'S REFERENCE, COPYRIGHT 1987, R. S. TARE, MCGRAW-HILL PUBLISHING, NEW YORK, NEW YORK.

TWO- AND THREE-DIMENSIONAL COMPUTATIONAL APPROACHES FOR STEEL PROJECTILES IMPACTING CONCRETE TARGETS, COPYRIGHT 1988, G.R. JOHNSON, R.A. STRYK, AND M.E. NIXON, EGLIN AFB, FLORIDA.

ENHANCEMENT AND INTEGRATION OF POST PROCESS UTILITIES FOR THE EPIC HYDROCODES

14-16

KENNETH GAGE
HIGH SCHOOL APPRENTICESHIP PROGRAM
AFATL/MNW
MENTOR: MICHAEL E NIXON
SUMMER 1990

OVERVIEW

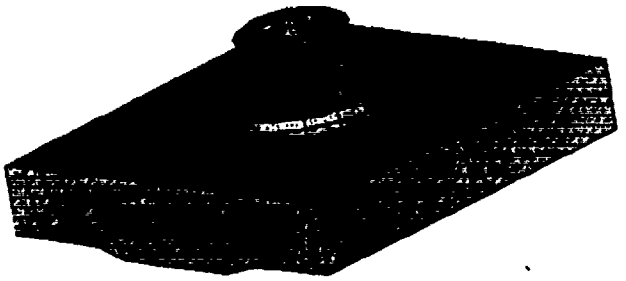
HYDROCODES AND UTILITIES

UTILITY ENHANCEMENTS

PHENIX INTEGRATOR

EPIC INPUT DATA DECKS





GRAPHICS PACKAGES

- ***PATRAN***
- ***GKS***

EPIC UTILITIES

EPIC2	EPIC3
TRAN2	EP3PAT
PATEP2	PATEP3
XYPLOT	TRANIX

EPIC TO PATRAN TRANSLATORS

- *EP3PAT*

  EPIC3-PATRAN TRANSLATOR

- *TRAN2*

  EPIC2-PATRAN TRANSLATOR

TRAN2

- REPLACES EP2PAT TRANSLATOR
- AUTOMATICALLY PRODUCES NEUTRAL FILE
- PRODUCES BOTH RESULTS FILES

PATRAN TO EPIC TRANSLATORS

- ***PATEP3***
  PATRAN-EPIC3 TRANSLATOR

- ***PATEP2***
  PATRAN-EPIC2 TRANSLATOR

XYPLOT

- EPIC2 DATA PLOTTER
- USES GKS
- CONVERTS DATA TO PATRAN FORMAT FOR USE WITH PPLOT

TRANIX

- DIVIDES ELEMENTS INTO FACES
- DETERMINES INTERIOR FACES
- REMOVES INTERIOR FACES
- REMOVES INTERIOR NODES
- CREATES NEW MODEL FILE OF JUST EXTERIOR FACES

TRANIX RESULTS

	NODES	ELEMENTS	DISK USAGE	PLOT TIME
ORIGINAL NEUTRAL MODEL FILE	15653	79824	18698 BLOCKS	OVER 3 HOURS
NEW NEUTRAL MODEL FILE	4418	8824	5038 BLOCKS	7 MIN, 45 SEC

PHENIX INTEGRATOR

- ABLE TO INITIATED THROUGHOUT SYSTEM
- CONTAINS A GENERAL OVERVIEW OF UTILITY
- CONTAINS STEP-BY-STEP INSTRUCTIONS FOR UTILITY
- PROVIDES EASY ACCESS TO ALL UTILITIES
- CAN BE EASILY MODIFIED TO INCORPORATE NEW UTILITIES DUE TO ITS HIERARCHAL STRUCTURE

```
#       Example Documentation Subprogram
#
#       Mike E Nixon
#       Computational Mechanics Team
#       Kenneth Gage
#       High School Apprentice
#       Summer 1990
#       AFATL/MNW
#
        clear
        more /user3/nixon/super.d/ep3pat.d/ep3pat.intro
        set ans = $<
        if ($ans == "1") then
        clear
        more /user3/nixon/super.d/ep3pat.d/ep3pat.info
        echo ""
        echo ""
        echo "Would you like to return to EPIC3-PATRAN documentation?"
        echo ""
        echo "Please type 'y' or hit RETURN."
        echo ""
        set a = $<
        if ($a == "y") then
        /user3/nixon/super.d/ep3pat.d/ep3pat.doc
        else if ($a == "") then
        /user3/nixon/super.d/ep3pat.d/ep3pat.doc
        endif
        else if ($ans == "2") then
        clear
        /user3/nixon/super.d/ep3pat.d/program.d/ep3pat
        else if ($ans == "3") then
        clear
        /user3/nixon/super.d/supershell
        else if ($ans == "4") then
        clear
        exit
        endif
```

RESULTS

- **REDUCED TRAINING TIME**
- **SMOOTHER TRANSITION**
- **INCREASED PRODUCTION**
- **VALUABLE EXPERIENCE**

EXPERIENCE GAINED

- FORTRAN
- GKS GRAPHICS
- UNIX COMMANDS
- EPIC INPUT DECK FORMAT

EQUIPMENT USED

- VAX 8650 MAINFRAME
- MULTIFLOW TRACE 14/200
- TEKTRONIX WORKSTATIONS
- VAXSTATION 3 GPX
- LNO3 LASER PRINTER
- KR VAXCLUSTER

ACKNOWLEDGEMENTS

MICHAEL E NIXON

HEATHER COX
FELLOW APPRENTICES
LORI PERILLO
BILL COOK
BIZHAN AREF & TEAS
AFATL/MNW

Design of In-House Radar Control and Data Acquisition Systems

AFATL/AGA/RF

Reid R. Harrison

School: Walton Sr. High

Mentor: Frank Arredondo

Summer 1990

Eglin A. F. B. Armament Laboratory

Air-to-Air Guidance Branch

1. Introduction and Background

This summer, I worked with Darryl Huddleston and Johnny Walker in the RF/MMW lab (room 484) of the Eglin Air Force Base Armament Laboratory (building 13). Mr. Huddleston is the program manager for the AGA/RF in-house radar program. The centerpiece of this program is an X-band pulse doppler radar system similar to the type found in F-16 fighters. The system consists of two antennas: a search antenna and a track antenna. The search antenna sweeps around 360 degrees and searches for targets in a large hemispherical domain. The track antenna is a parabolic dish dual-feed-horn antenna that can move to a specific azimuth and elevation to track individual targets. Both antennas are located on the roof of the building 13 annex, directly above the RF/MMW lab. The short-term goal of this program is to control this radar system with a T-248 microcomputer. Capt. Jeff Barnes is using Microsoft C 5.1 to create radar control software. Much of the work I did this summer involved writing special-purpose additions to his main control program, 'bc.c'. I also wrote some independent C software.

2. Description of Research

Objective

During this summer, I was assigned the task of designing and implementing an automated, computer-controlled system to map radar antenna patterns. The following is a list of equipment that I used as a part of this system:

- AGA/AGS X-band pulse doppler in-house radar
- I-248 microcomputer with:
    - 1553 bus controller card
    - GP488B HP-IB bus interface card
    - Microsoft C 5.1
    - Borland Turbo C
- HP 8757C scalar network analyzer
- HP sweep oscillator with:
    - HP 2.0 - 20.0 GHz insert
- X-band standard gain antenna horn mounted on tripod
    (19.7 cm x 14.6 cm rectangular aperture)
- X-band standard gain antenna horn
    (7.6 cm x 5.7 cm rectangular aperture)
- RF amplifier

Track Antenna Control

The first requirement for the antenna-mapping system was control of the track antenna. I studied the 1553 codes associated with antenna movement and position. To move the

track antenna, F03 words 07-09 must be supplied with the X, Y, and Z coordinates of a target (in meters). Since I wanted to move the antenna only to specific azimuths and elevations, I wrote a C routine (rae_to_xyz()) to convert azimuth and elevation to X, Y, and Z coordinates in radar format. At first, I thought that any range could be specified for the computations. However, the radar only works properly when 16384 meters (maximum range) is used in the conversion. The reason for this is unknown.

After the conversion routine was established, I wrote movedish(), a C routine to move the track antenna. I later created a modification of movedish() called movewait(). This routine moves the track antenna and does not return to normal program execution until both the elevation and azimuth sensors return values less than 0.3 degrees from the requested values, or until 3.0 seconds have passed. This routine utilizes another routine I wrote, ant_coord(), which returns the value of either the azimuth or elevation position sensor as a floating point number. These values are read from R01 words 04-07.

HP-IB control

In order to automate serval pieces of lab equipment, I installed a GP488B HP-IB interface card in the Z-248 computer. I installed the software driver on hard disk, and created a

text file called 'config.txt' in the ieee488 subdirectory of drive C. This file describes how the DIP switches and jumpers were set on the card prior to installation. I then learned how to control an HP plotter and power meter from Turbo C. Finally, I controlled the HP scalar network analyzer and sweep oscillator from Microsoft C. In order to send commands to the sweep oscillator, which is not directly connected to the Z-248, a passthru address (17) in the network analyzer must be told the oscillator's address (19). I also learned how to read data from the network analyzer and save it to a file on hard disk.

Physical Set-up

I mounted the transmit antenna horn on a camera mount on the track antenna. This antenna was connected via RF cable to the output of the sweep oscillator. The receive horn was placed on a tripod at the end of a waveguide extension. This placed the horns about 35 feet apart. The waveguide was connnected via RF cable to an RF amplifier, which fed into the 'A' input in the scalar network analyzer. The Z-248 computer was connected to the radar over the 1553 bus. A standard HP-IB cable connected the scalar network analyzer and the GP488B card in the Z-248.

The Completed Program

The main antenna mapping routine is accessed from bc2.c

by pressing the 'R' key. The brings up requesters that allow the user to input the starting and stopping azimuths and elevations. Also, the program requests the increment size for azimuth and elevation scans. After this information is entered, the program presets the HP equipment, sets a frequency range of 9.6 - 10.0 GHz, and sets the power level to 10.0 dBm. The analyzer's cursor is turned on and positioned at 9.8 GHz. Next, the program turns the RF on and begins to move the track antenna step by step, reading and recording the power level at each stop. The data is saved to the file 'h:datafile.rad'. The format of this file is described in the technical documentation. The routine sendfloat() is used to write floating point numbers to 4-byte fields in a disk file. The scan rate is about 2 readings per second. A complete scan (-60 to +30 azimuth, -7 to +85 elevation) in 1 degree increments takes about 85 minutes. After the scan is complete, the RF is turned off, the file is closed, and the track antenna is returned to 0 degrees azimuth, 0 degrees elevation.

Graphics Display Routines

Using Turbo C on a Z-248, I wrote programs that displayed the radar antenna pattern data files as two- and three-dimensional plots. Different colors were used to denote ranges of power levels. In one version, the power levels

above 3 dBm below the maximum value are colored yellow. Therefore, the beamwidth is visible. These graphics routines were eventually incorporated into 'antplot.c', a program written in Turbo C. This program displays full size plots. Full size means a scan from azimuth -60 to +30 and elevation -7 to +85 in 1 degree steps. Plots can be displayed in two-dimensional, three-dimensional, or spherical plots. Two coloring schemes are available. One distributes twelve colors evenly from the maximum power level to the minimum. The other shows the antenna beamwidth.

3. Results

Two antennas were observed: a standard gain X-band antenna horn, and the track antenna, a parabolic dish with a three-foot diameter. The standard gain horn exhibited a beamwidth very close to the width predicted by equations. Also visible in the plot were strange patterns that were eventually attributed to multipathing, or reflections off of clutter on the roof. The track antenna plot showed a very narrow beamwidth, with vertically-positioned sidelobes. Also visible in the plot was the dual-feed-horn nature of the antenna.

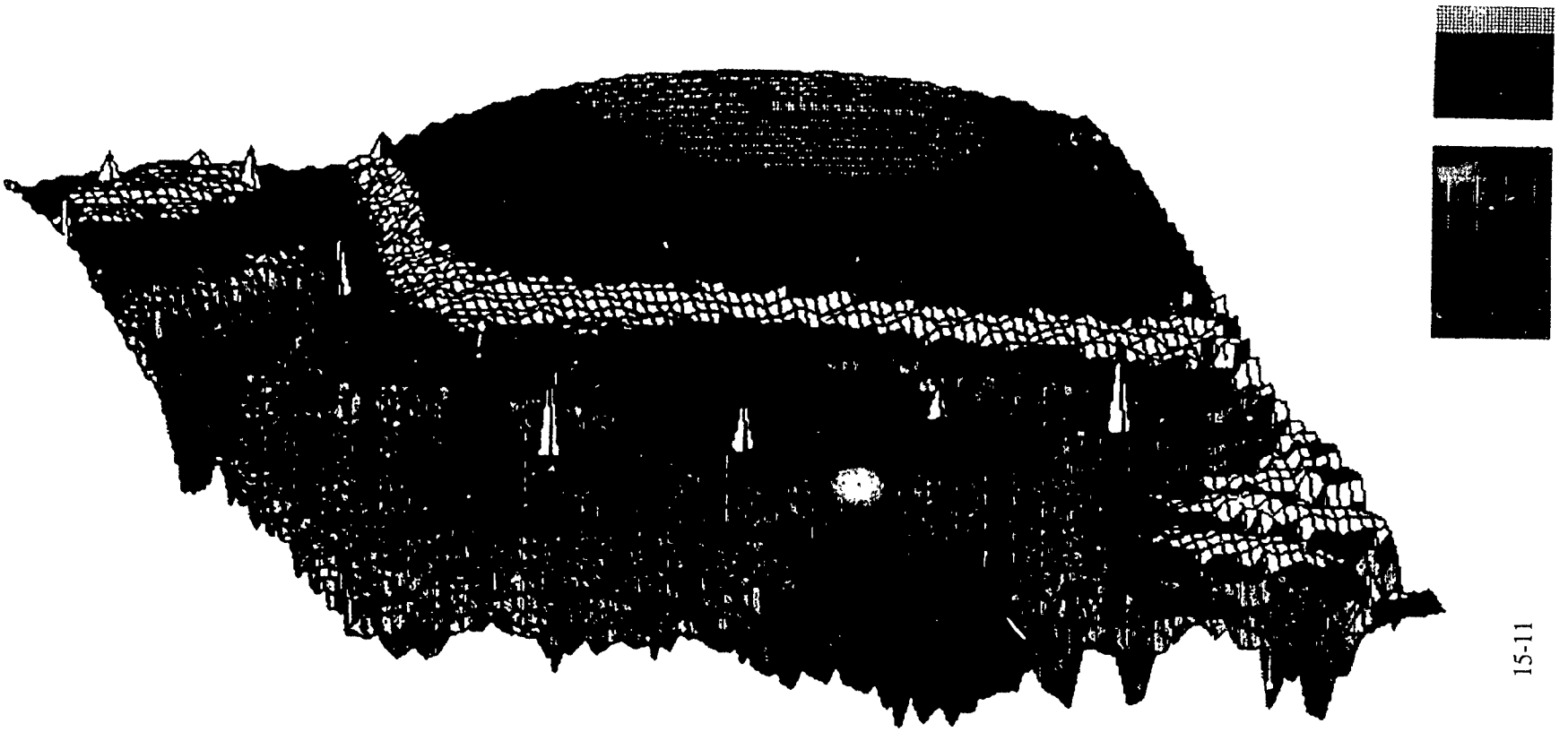




Each Color Band Covers 4.379084 dBm
Number of Data Pts : 8463
Minimum Azimuth : -61.501465 degrees
Maximum Azimuth : 30.267334 degrees
Minimum Elevation : -7.053223 degrees
Maximum Elevation : 85.001221 degrees
Minimum Power Level: -69.371002 dBm
Maximum Power Level: -15.822000 dBm



Each Color Band Covers 4.379084 dBm
Number of Data Pts : 8463
Minimum Azimuth : -61.501465 degrees
Maximum Azimuth : 30.267334 degrees
Minimum Elevation : -7.053223 degrees
Maximum Elevation : 85.001221 degrees
Minimum Power Level: -68.371002 dBm
Maximum Power Level: -15.822000 dBm



Each Color Band Covers 4.379084 dBm
Number of Data Pts : 8463
Minimum Azimuth : -61.501465 degrees
Maximum Azimuth : 30.267334 degrees
Minimum Elevation : -7.053223 degrees
Maximum Elevation : 85.001221 degrees
Minimum Power Level: -68.371002 dBm
Maximum Power Level: -15.822000 dBm

4. Acknowledgements

I would like to thank the following people for their continued support during this summer.

Mr. Frank Arredondo

Mr. Darryl Huddleston

Capt. Jeff Barnes

Mr. John Walker

Maj. Bob Johnson

Mr. Robert Brown

Mr. Tom Speed

Mr. Bruce Quarles

Mr. David Rogers

Ms. Christine Riendeau

5. Technical Documentation

bc2.c

The file 'bc2.c' is my modified version of Capt. Barnes' 'bc.c'. It includes serveral new functions and two new keypress routines ('A' and 'R'). I removed the 'M' keypress option (modify F03 message mask) to reduce compilation time.

New Fuctions

float ant_coord(int which_one, struct link_STR *blink)

Purpose:

Returns the value of either the azimuth or elevation position sensor of the track antenna.

Usage:

The variable 'which_one' should be either 'AZIMUTH' or 'ELEVATION', which are #defined as 5 and 7, respectively. The struct should always be as it appears in the example. This routine reads R01 words 04/05 (azimuth) or 06/07 (elevation).

Example:

```
float az;
az = ant_coord(AZIMUTH,
        &link_bf_1553.link[R01_buffer.link_index]);
```

void disp_angle(int vert, int horz, struct link_STR *blink)

Purpose:

Display the track antenna position sensor readings (azimuth and elevation) on the graphics screen.

Usage:

The variables 'vert' and 'horz' position the text on the screen. The struct should always be as it appears in the example.

Example:

```
disp_angle(19, 41,
           &link_bf_1553.link[R01_buffer.link_index]);
```

void rae_to_xyz(double rae[3], WORD xyz2[3])

Purpose:

Convert range, azimuth, and elevation to X, Y, and Z coordinates in F03 words 07-09 format.

Usage:

The array 'rae' is range (rae[0]), azimuth (rae[1]), and elevation (rae[2]). For track antenna movement, range should always be 16384.0. Azimuth and elevation are given in degrees. The array 'xyz2' is X, Y, and Z coordinates,

repectively. Note: This function modifies the values of 'xyz2' AND 'rae'.

Example:

```
float spherical[3] = { 19384.0, -30.0, 45.0 };
WORD data_for_1553[3];
rae_to_xyz(spherical, data_for_1553);
```

void timedelay(double seconds)

Purpose:

Pause for a while.

Usage:

The variable 'seconds' is the number of seconds that the program does nothing but execute pc1553_loop().

Example:

```
timedelay(3.0);      /* pause three seconds */
```

void movedish(float azimuth, float elevation)

Purpose:

Move the track antenna to a specific azimuth and elevation.

Usage:

The variables 'azimuth' and 'elevation' are given in degrees. The azimuth range is -60 degrees to +30 degrees. The elevation range is -7.5 degrees to +85 degrees.

Example:

```
movedish(0.0, 30.0);
```

int movewait(float azimuth, float elevation)

Purpose:

Move the track antenna, as in movedish(), but do not resume normal program operation until both position sensors return values less than 0.3 degrees from the requested values, or 3.0 seconds elapse.

Usage:

Same as movedish(), except that movedish() will not attempt to move the antenna if the azimuth and/or elevation variables are not within the specified ranges. Instead, the function returns a -1.

Example:

```
movewait(0.0, 30.0);
```

void send_float(FILE *out, float number)

Purpose:

Write a float to a disk file in a 4-byte (most compact) format.

Usage:

The variable 'number' is written to the file pointed to by 'out'.

Example:

```
FILE *myfile;
/* open file here */
sendfloat(out, 3.1416);
```

Antenna Pattern File Specifications

When a measurement is made by bc2.c, the resulting data is saved to h:datafile.rad. The format of this data file is as follows:

(1) five I.D. characters: 'DIVAD'

(2) a 4-byte float telling how many data segments follow.

Each data segment consists of:

(a) azimuth sensor reading (4-byte float)

(b) elevation sensor reading (4-byte float)

(c) power level in dBm (4-byte float)

6. References

In addition to the expertise of my mentor and other people in my branch and in the lab, I used the following resources:

Borland Turbo C documentation

Microsoft C 5.1 documentation

Hughes Radar Handbook

IOtech GP488B documentation

Appendices can be obtained from
UNIVERSAL ENERGY SYSTEMS, INC.

1990 HSAP Final Event Summary
SAI / AFATL

by

Derek Holland

MENTOR : Starla Christakos

Acknowledgements

Starla Christakos - For her patience with me.

Troy Urquhart - For helping in all things.

Jeff Brooks -
Paul McCarley -] For help in the C30 sim.

Alan Andrews - For his "C" expertise.
Charles Coker - For help with any SUN problems.
Mike Couvillon - For something or other.
TJ & Matt - For help from "Co-Op Corner."
Bob LeBeau - For his insanity.
Dot Sauer - For accidentally donating her terminal.
Dorothy Williams - For her printer generosity.
Everybody else at SAI

Fellow HSAP's - Never letting me get TOO serious.
Mr. Harrison - For giving me a chance to work here.

Summary Order:

Background and Definitions

Projects:

I. Smart PreImage Selector

II. MultiImaging

III. C30 Simulator & Bridgeboard

IV. Miscellaneous

Conclusion

Background and Definitions

This past summer I worked for the Air Force Armament Technology Lab, located on Eglin Air Force Base, as a participant in Universal Energy System's High School Apprenticeship Program. During that time I researched and studied under Starla Christakos, chief of the elite Signal Processing section, Guided Interceptor Technology / Strategic Defense Initiative.

I was involved in several projects over the course of the summer, most of which involving direct creative programming, as opposed to straitlaced punching and batching. These are the "adventures" that I will describe in this report, as I deem them worthy of your ears.

The first of these projects involved writing a smart algorithm, which, when activated, would scan an image and choose its optimum processing path. The second was "like unto the first" only in its base - here I wrote the graphics code and Signal Processing for a new simulator - one that was multi-imaging and capable of processing an entire endgame scenario. The third (and also the most boring, I might add), involved direct programming of Signal Processing chips, through a special C compiler. This was done the expedite the entire guidance process for the interceptor. (NOTE: I also assisted in many other projects my colleague and fellow HSAP, Troy Urquhart; however, not wishing to intrude upon his report, said projects will be dealt with in a special MISCELLANEOUS section).

Since all projects were separate and unique, I have treated them as such in this report. For the unfortunate soul who wishes short, concise definitions of what I mean (example: "What the *#%* is that HSAP talking about?!"), following this introduction is a series of BASIC terms that I shall endeavor to use correctly.

Not to seem spoiled here, I must say that I have enjoyed this summer a great deal. Learning skills never taught in a classroom, I feel now better-equipped to proceed to college, well that is, after I complete my next summer apprenticeship in 1991.

DSM

The Digital Seeker Model is a simulation which runs on a SUN workstation 3, and is blessed the knowledge that it served as the prototype for my smart selector. It consists of two main processes, one a controller C graphics interface, and the other a number-crunching FORTRAN module.

Since I worked with both, I feel you are entitled to a brief summary of their components.

On the FORTRAN side, in ordinary flow, a user-defined set of qualifications on the integral parts of an interceptor is fed through a course of computer actuated designs. A user can then "step through" each part of the interceptor's Signal Processing and Seeker many stages and functions. I refuse here (quite adamantly, I must say) to ramble on about line such-and-such and what IT does. For the ultimate programmer, I have included a copy of my modified simulator.

The C part mainly displays the FORTRAN output. Not really much to say, and I refuse to dump the entire graphics cluster, for I didn't write this part, I only modified it to make it faster.

MISP

MISP stands for Multiple Image Signal Processor. This is my program (well, I have to share credit with the other apprentice here), and so a listing is included. Basically, MISP runs as DSM, save the fact that MISP allows the user to view an entire flight path of an interceptor. He/she is NOT limited to one image at a time, as in DSM!! (I was famous for my 15 minutes for this one, Mr. Warhol) This simulation is written entirely in C, for it is the language of champions, and I shall continue as these bits of humor are probably not FUNNY.

C30sim

C30sim was not the only name given to this piece of wonder. Truly fascinating vulgarities and expletives described this package which allowed one to DIRECTLY program the chips of an interceptor, specifically the ones devoted to signal processing. Not really much more to say here.

SIGPROC Ops.

Whenever I speak of different Signal Processing specifics, I shall make quick reference to a particular routine. The following is a list of definitions for the technically weak-spirited:

Kinetic Energy Weapon - What I work with. Specifically, an SDI vehicle, which, residing in space, is launched at hypersonic velocities at oncoming ICBM's.

Threshold - The process of systematically cutting out noise in a picture, usually with a fixed value or some linear equation.

Edge Operator - A small matrix , or system of matrices that is scanned over an image to produce a desired shift in the intensity values around some edge phenomenon.

Transform - An operation of changing the format or general characteristics on an image for storage or analysis purposes (Ex: Fourier transforms are used to analyze discrete spectra of particular images)

Kerse - The engineer's term for any vulgar expletive, thus the one used in this report.

Slave - the convenient term for an HSAP in my office (This is a joke. My office is very loving)

Rather than tell more stupid jokes, I think we'll move along.

WELL, TURN THE PAGE...

Figures Listed

1. Generic Interceptor

2. Interceptor Closeup

3. MISP flow

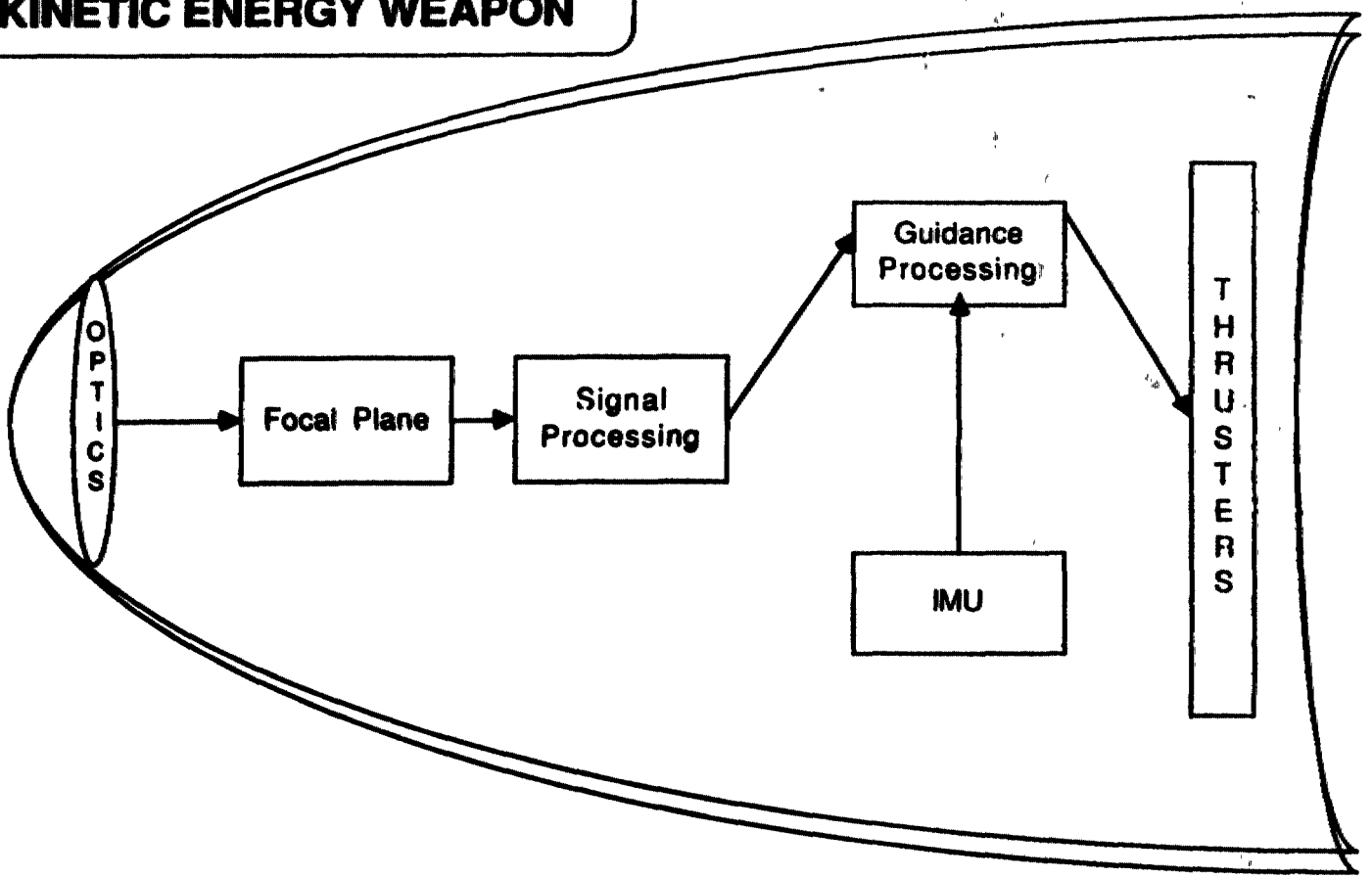
4. MISP Signal Processing

5. Smart Selector Definition

6. Smart Selector Flow

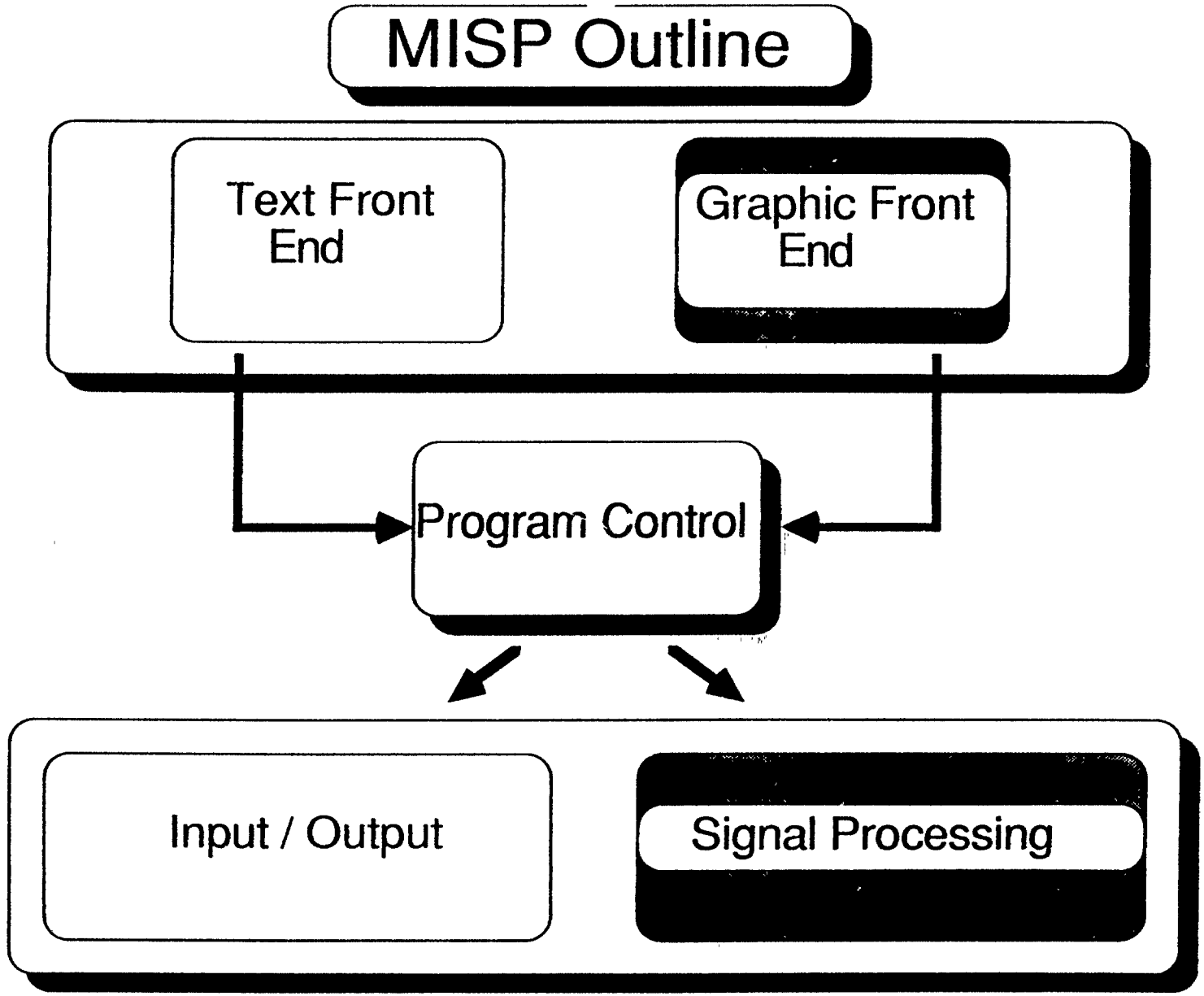
7. Guidance Explanation

KINETIC ENERGY WEAPON



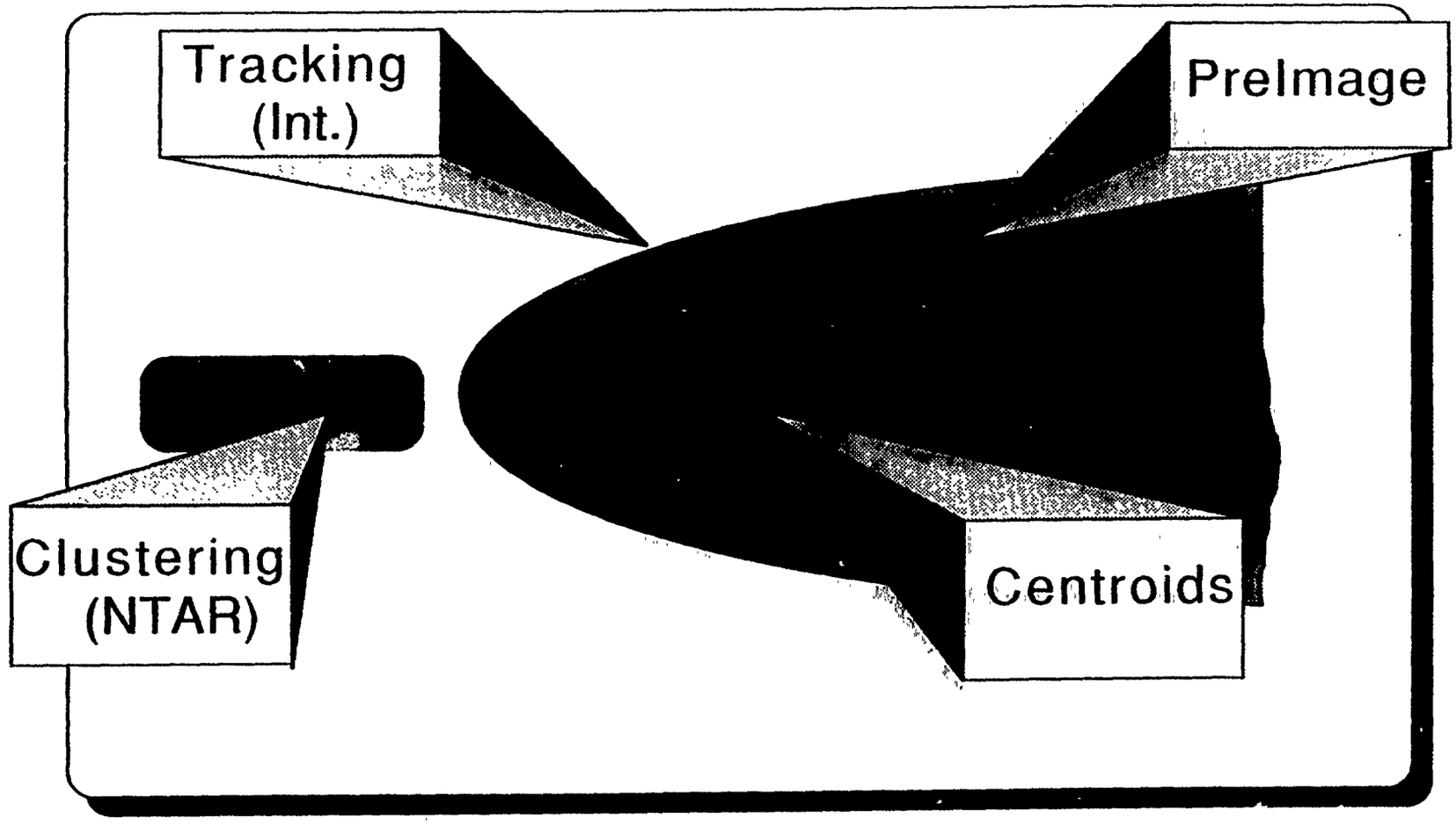
Signal Processing - Extraction of relevant data from irrelevant noise or interference.

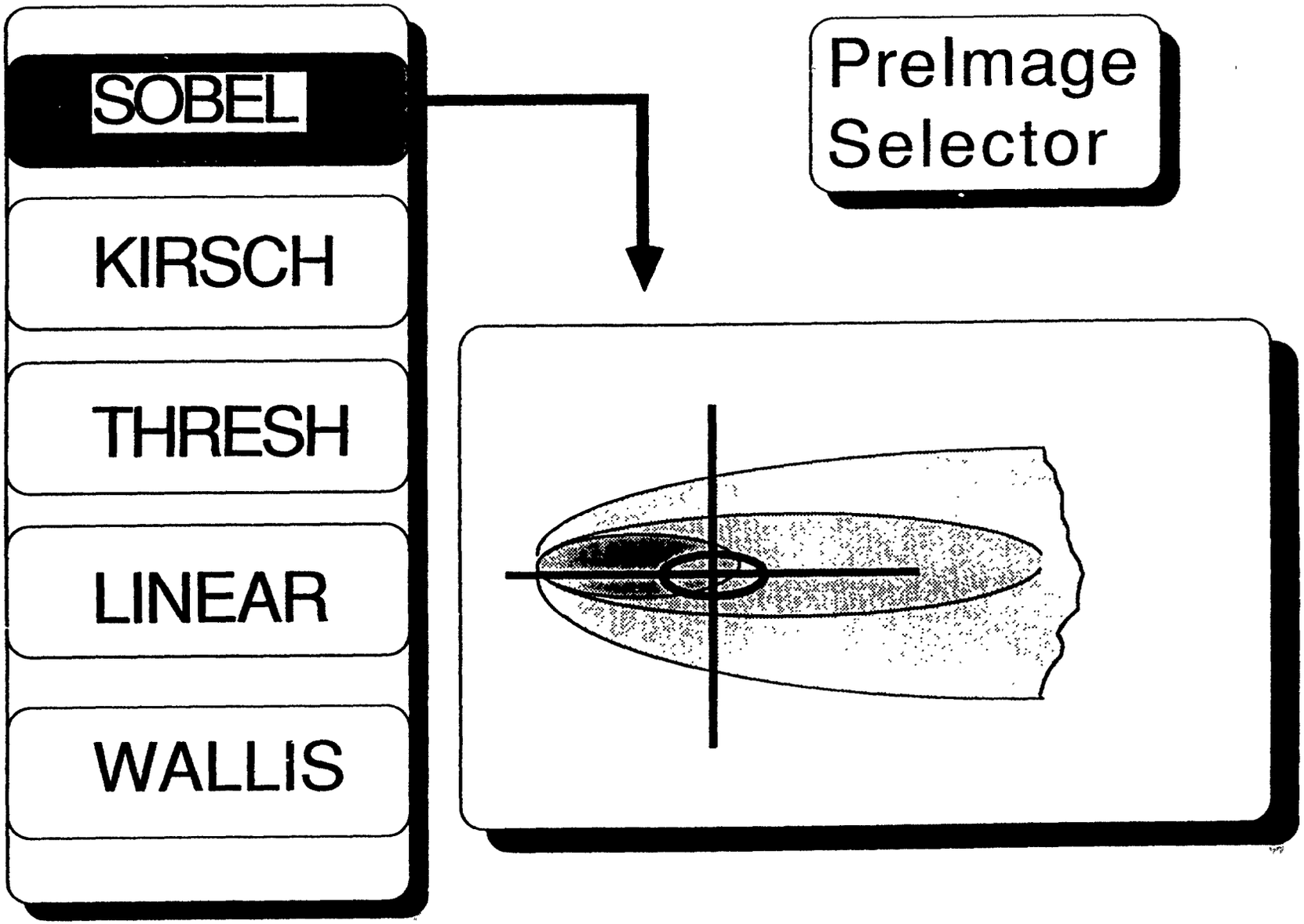
Interceptor
Guidance
Guidance Laws
Kalman Filtering
Autopilot Comp.
Chip Sharing
Signal Processing
Algorithm Processing
Target Tracking
Target Output


MISP Outline
Text Front End
Graphic Front End
Program Control
Input / Output
Signal Processing

Main Signal Processing Operations

MISP



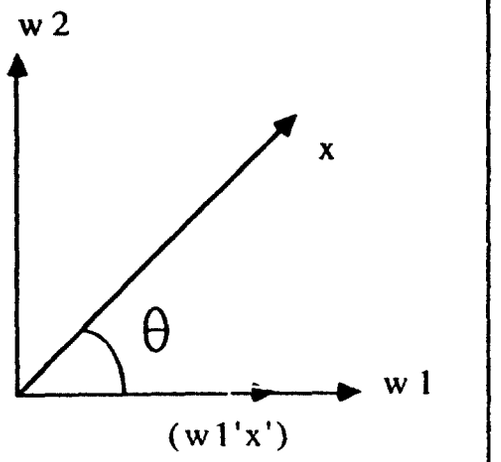

SOBEL
KIRSCH
THRESH
LINEAR
WALLIS
PreImage
Selector

Selector Flow

1

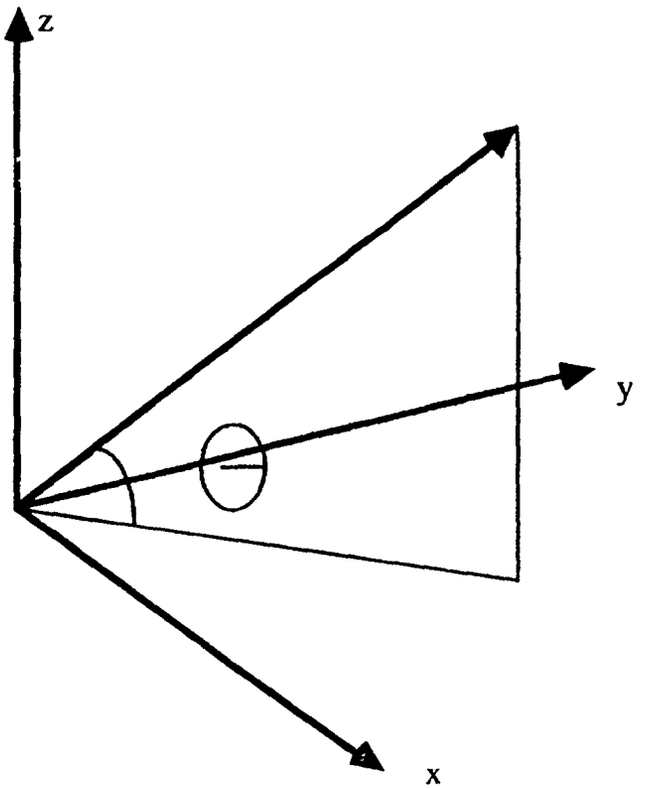
3	1	1
1	3	1
1	1	3

2

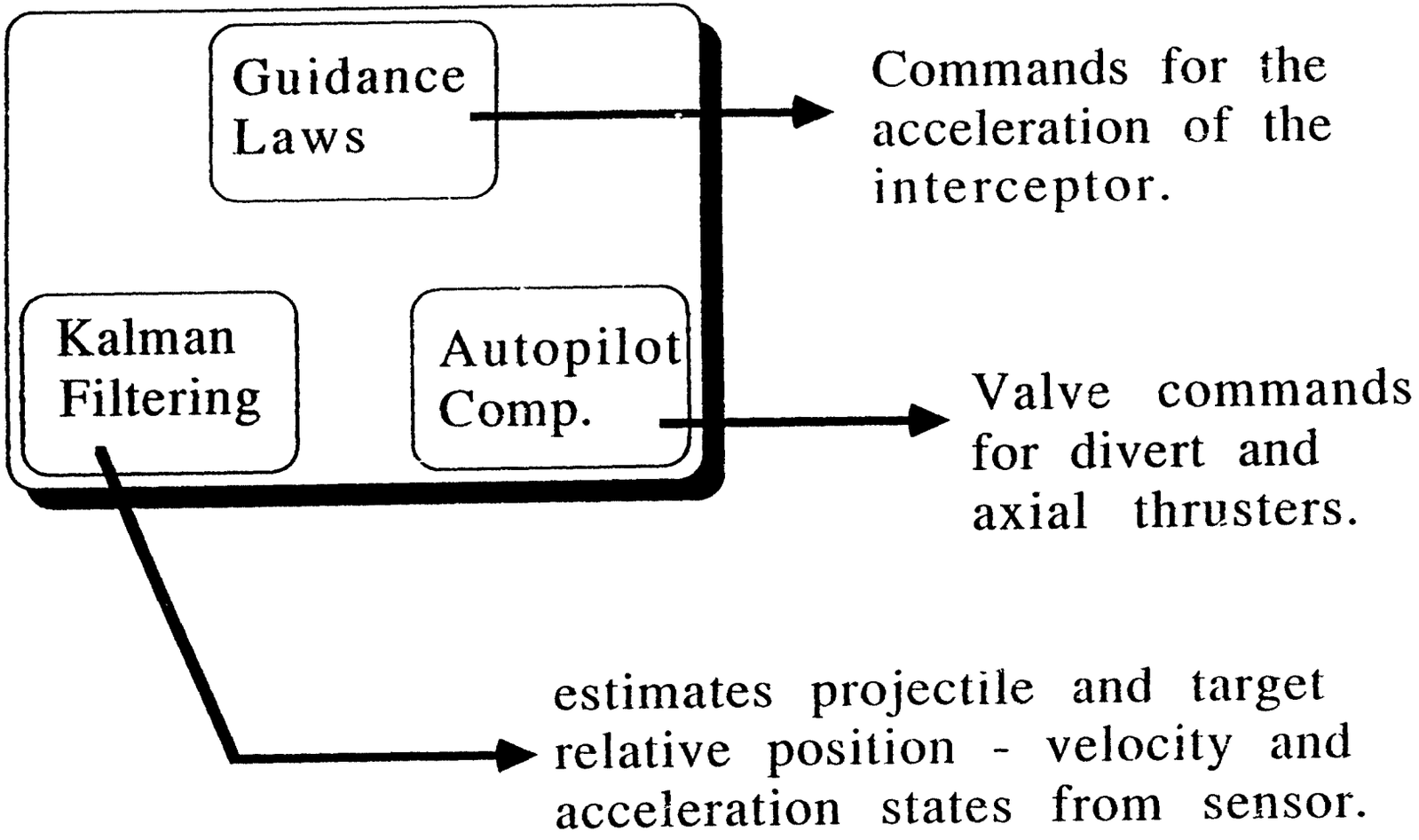


$w1'x' = \|w1\| \|x\| \cos \theta$

3



Guidance Summary



Projects

I. Smart PreImage Selector

OBJECTIVE

Funny how ideas start. When I first arrived, trying to reacquaint myself with the complexity of the Lab, I was playing around with the Digital Seeker Model, relearning its "wonders." I was suddenly struck with several thoughts. In a non-user simulation, or even in a real interceptor, for that matter, no time will be available to choose between algorithms in a leisurely manner. Decisions must be made quickly. It was then that I realized also that no one had formulated a simple PURELY MATHEMATICAL way of choosing which way to processing any given image (for they are all different - no ONE way works). Since, like I said, no one had ever tried this line of research before, I developed lots of "fans." I then set out to design an algorithm myself that, when given a library of possible routines, and any image, could match the best routine to the image.

THEORY

I researched matrices and vectors and all that for a long while, trying to find some key to the mess of pure-math differentiation. Since I wanted to stay away from AI or neural-networking (the chips were not designed for that), I ran into many "brick-walls" of measurement. For one, how would it "know" the characteristics of each image in relation to other data? Was there any "signature" for masks to identify?

NO. I chose then to identify the mask characteristics of a particular algorithm by subspace axes. Therefore, to calculate the best mask, for any number of them, one "pushes" some over an image, and maps their gradient vector or "response" onto the multidimensional subspace (number of dimensions == number of masks). The best mask to run will be the one with the smallest angle (theta) from the vector to the axis.

This year, I put in eight linear/directional compass methods and tested sensitivity to detect edges. Results were very satisfactory. The program always chose the mask that I would choose. Whether that was the BEST mask remains to be seen (JOKING).

Enough of theory. As you can see, it's pretty easy, and MOSTLY mine. I must admit, the subspace idea is not.

PROCEDURE

I wrote it. What more can I say???

RESULTS

It worked - I mean really worked! Again, I felt like a million bucks. Wow. Maybe next year I'll even optimize it.

BUGS

Increasing rapidly as one adds on further algorithms, is CPU time. NOTE: This method only works with libraries of 25 or fewer masks. Any more slows down the overall computation time of the interceptor.

REFLECTIONS

I will never attempt something like this ever again.
(Derek hath spoken)

Digital Seeker Model (DSM)

excerpt:

ns92sig.f
(signal process library)

```
c------\SS2sig.f
c-Program name et a   operators
c----Dere. ho  and

      Subroutine extsigpt(pfpa fsize,sigtype
      character*80 sigtype,cr
      real*4 pfpa(128,128)
      integer fsize,signum,unk

 10   format(a80)
      unk=0
      if (sigtype .eq. 'SIG_') then
         unk = 1
 20      print *,'Enter type of Signal Procesing'
         print *,' Types available = '
         print *,'   - SIG_SOBEL'
         print *,'   - SIG_KIRSCH'
         print *,'   - SIG_GENERIC2X2'
         print *,'   - SIG_ROBERTS1'
         print *,'   - SIG_ROBERTS2'
         print *,'   - SIG_WALLIS note. use linear scaling
         print *,'   - SIG_WEIRDWALLIS note. use linear scaling'
         print *,'   - SIG_LAPLACIAN1'
         print *,'   - SIG_LAPLACIAN2'
         print *,'   - SIG_LAPLACIAN3'
         print *,    - SIG_COMPASS.N'
         print *,'   - SIG_COMPASS.NE'
         print *,'   - SIG_COMPASS.E
         print *,'   - SIG_COMPASS.SE'
         print *,'   - SIG_COMPASS.S'
         print *,'   - SIG_COMPASS.SW'
         print *,'   - SIG_COMPASS.W
         print *,'   - SIG_COMPASS.NW'
         print *,'   - SIG_HORIZONTAL1'
         print *,'   - SIG_HORIZONTAL2'
         print *,'   - SIG_VERTICAL1'
         print *,'   - SIG_VERTICAL2'
         print *,'   - SIG_HISTOGRAM_EQ'
         print *,'   - SIG_HOUGHTRANS'
         read(*,10) sigtype
         signum = 1
      endif

      if(sigtype eq 'SIG_SOBEL') then
          call sobel(pfpa,fsize)
          signum = 2
          elseif(sigtype eq.'SIG_GENERIC2X2') then
             call gener c(pfpa,fsize)
          signum = 3
          elseif(sigtype.eq 'SIG_ROBERTS1') then
             ca.l roberts1(pfpa,fsize)
          signum = 4
          elseif(sigtype.eq.'SIG_WALLIS') then
             call wallis(pfpa,fsize)
          signum = 5
          elseif(sigtype eq.'SIG_WEIRDWALLIS') then
             call weirdwallis(pfpa,fsize)
          signum = 6
          elseif(sigtype eq 'SIG_LAPLACIAN1') then
             call laplacian1(pfpa,fsize)
          signum = 7
          elseif(sigtype.eq 'SIG_LAPLACIAN2') then
             call laplacian2(pfpa,fsize)
          signum = 8
          elseif(sigtype.eq 'SIG_LAPLACIAN3') then
             call laplacian3(pfpa,fsize)
          signum = 9
          elseif(sigtype.eq 'SIG_COMPASS.N') then
             call compassn(pfpa,fsize)
          signum = 10
          elseif(sigtype.eq.'SIG_COMPASS.NE') then
             call compassne(pfpa,fsize)
          signum = 11
          elseif(sigtype.eq.'SIG_COMPASS.E') then
             call compasse(pfpa,fsize)
          signum = 12
          elseif(sigtype eq.'SIG_COMPASS.SE') then
             call compassse(pfpa,fsize)
          signum = 13
          elseif(sigtype.eq.'SIG_COMPASS.S') then
             call compasss(pfpa,fsize)
          signum = 14
          elseif(sigtype.eq.'SIG_COMPASS.SW') then
             call compasssw(pfpa,fsize)
          signum = 15
          elseif(sigtype.eq.'SIG_COMPASS.W') then
             call compassw(pfpa,fsize)
          signum = 16
```

```
      subroutine generic(f,fsize
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'Generic 2 by 2
      do 20 i = 2,fsize-1
         do 10 j = 2,fsize-1
            a = (f(i,j)-f(i+1,j))
            b = (f(i,j)-f(i,j+1))
            buff(i,j) = sqrt(a*a+b*b)
   10    continue
   20 continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
   30    continue
   40 continue

      end
c-------------------------------------------------
c------Roberts1 Operator
c------Digital Image Processing
c------p  177
c------Derek Holland

      subroutine roberts1(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'Roberts1 Operator
      do 20 i = 2,fsize-1
         do 10 j = 2,fsize-1
            a = (f(i,j)-f(i+1,j+1))
            b = (f(i+1,j)-f(i,j+1))
            buff(i,j) = sqrt(a*a+b*b)
   10    continue
   20 continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
         f(i,j) = buff(i,j)
   30    continue
   40 continue

      end
c-------------------------------------------------
c---------Wallis Operator
c---------Digital Image Processing
c---------p  489
c---------Derek Holland

      subroutine wallis(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'Wallis Operator '
      do 20 i = 2,fsize-1
         do 10 j = 2,fsize-1
50          x = alog10(f(i,j))
            y = alog10(f(i,j+1))/4
            z = alog10(f(i+1,j))/4
            d = alog10(f(i,j-1))/4
            h = alog10(f(i-1,j))/4
            buff(i,j) = x - y - z - d - h
10       continue
20    continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
         f(i,j) = buff(i,j)
30       continue
40    continue
      end
c-------------------------------------------------
c---------WEIRDWallis Operator
c---------Digital Image Processing
c---------p. 489
c---------Derek Holland

      subroutine weirdwallis(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
```

```
20    continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
30       continue
40    continue

      end
c-----------------------------------------------------
c----------Compass Direction Mask = SOUTHEAST
c----------DIGITAL IMAGE PROCESSING
c----------p. 481
c----------DH

      subroutine compassse(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'SOUTHEAST.'
      do 20 i = 2,fsize-1
         do 10 j= 2,fsize-1
            buff(i,j) = -f(i-1,j+1)-f(i,j+1)-f(i+1,j+1)-
     +       f(i-1,j)-2*f(i,j)+f(i+1,j)+f(i-1,j-1)+f(i,j-1)+
     +       f(i+1,j-1)
10       continue
20    continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
30       continue
40    continue

      end
c-----------------------------------------------------
c-------Compass direction mask - SOUTH
c-------Digital Image Processing
c-------p. 481
c-------DH

      subroutine compasss(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'SOUTH '
      do 20 i = 2,fsize-1
         do 10 j= 2,fsize-1
            buff(i,j) = -f(i-1,j+1)-f(i,j+1)-f(i+1,j+1)-
     +       f(i-1,j)-2*f(i,j)+f(i+1,j)+f(i-1,j-1)+f(i,j-1)+
     +       f(i+1,j-1)
10       continue
20    continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
30       continue
40    continue

      end
c-----------------------------------------------------
c-------Compass direction mask - SOUTHWEST
c-------Digital Image Processing
c-------p. 481
c-------DH

      subroutine compasssw(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'SOUTHWEST.'
      do 20 i = 2,fsize-1
         do 10 j=2,fsize-1
            buff(i,j) = f(i-1,j+1)-f(i,j+1)-f(i+1,j+1)+
     +       f(i-1,j)-2*f(i,j)-f(i+1,j)+f(i-1,j-1)+f(i,j-1)+
     +       f(i+1,j-1)
10       continue
20    continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
30       continue
40    continue
```

```
      end
c-------------------------------------------------
c-------Compass direction mask - WEST
c-------Digital Image Processing
c-------p  481
c-------DH

      subroutine compassw(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'WEST '
      do 20 i = 2,fsize-1
         do 10 j=2,fsize-1
            buff(i,j) = f(i-1,j+1)+f(i,j+1)-f(i+1,j+1)+
     +       f(i-1,j)-2*f(i,j)-f(i+1,j)+f(i-1,j-1)+f(i,j-1)-
     +       f(i+1,j-1)
10       continue
20    continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
30       continue
40    continue

      end
c-------------------------------------------------------
c------Compass direction mask - NORTHWEST
c------Digital Image Processing
c------P  481
c------DH

      subroutine compassnw(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'NORTHWEST OPERATOR.'
      do 20 i = 2,fsize-1
         do 10 j =2,fsize,1
            buff(i,j) = f(i-1,j+1)+f(i,j+1)+f(i+1,j+1)+
     +       f(i-1,j)-2*f(i,j)-f(i+1,j)+f(i-1,j-1)-f(i,j-1)-
     +       f(i+1,j-1)
10       continue
20    continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
30       continue
40    continue

      end
c-------------------------------------------------------
c----------Horizontal linear detection mask
c----------Digital Image Processing
c----------p  481
c----------DH

      subroutine horizontal1(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'Horizontal.'
      do 20 i = 2,fsize-1
         do 10 j = 2,fsize-1
            buff(i,j) = f(i,j)-f(i,j+1)
10       continue
20    continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
30       continue
40    continue

      end
c-------------------------------------------------------
c-------Vertical linear detection mask
c-------Digital image processing
c-------p  480
c-------DH

      subroutine vertical1(f,fsize)
```

```
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
30       continue
40    continue

      end
c-----------------------------------------------------
c-------Kirsch Edge Operator
c-------Digital Image Processing
c-------p  489
c-------DH

      subroutine kirsch(f,fsize)
      real*4 f(128,128)
      integer fsize,i,j
      real*4 buff(128,128)
      common /buffers/ buff
      print*,'Kirsch '
      do 20 i = 2,fsize-1
         do 10 j = 2,fsize-1
            a = 5*(f(i-1,j+1)+f(i,j+1)+f(i+1,j+1))-3*(f(i+1,j)+
     +          f(i+1,j-1)+f(i,j-1)+f(i-1,j-1)+f(i-1,j))
            b = 5*(f(i,j+1)+f(i+1,j+1)+f(i+1,j))-3*(f(i+1,j-1)+
     +          f(i,j-1)+f(i-1,j-1)+f(i-1,j)+f(i-1,j+1))
            c = 5*(f(i+1,j+1)+f(i+1,j)+f(i+1,j-1))-3*(f(i,j-1)+
     +          f(i-1,j-1)+f(i-1,j)+f(i-1,j+1)+f(i,j+1))
            d = 5*(f(i+1,j)+f(i+1,j-1)+f(i,j-1))-3*(f(i-1,j-1)+
     +          f(i-1,j)+f(i-1,j+1)+f(i,j+1)+f(i+1,j+1))
            e = 5*(f(i+1,j-1)+f(i,j-1)+f(i-1,j-1))-3*(f(i-1,j)+
     +          f(i-1,j+1)+f(i,j+1)+f(i+1,j+1)+f(i+1,j))
            t = 5*(f(i,j-1)+f(i-1,j-1)+f(i-1,j))-3*(f(i-1,j+1)+
     +          f(i,j+1)+f(i+1,j+1)+f(i+1,j)+f(i+1,j-1))
            g = 5*(f(i-1,j-1)+f(i-1,j)+f(i-1,j-1))-3*(f(i,j+1)+
     +          f(i+1,j+1)+f(i+1,j)+f(i+1,j-1)+f(i,j-1))
            h = 5*(f(i-1,j)+f(i-1,j+1)+f(i,j+1))-3*(f(i+1,j+1)+
     +          f(i+1,j)+f(i+1,j-1)+f(i,j-1)+f(i-1,j-1))
            x = max(abs(a),abs(b),abs(c),abs(d),abs(e),abs(t),
     +          abs(g),abs(h))
            buff(i,j) = max(1 ,x)
10       continue
20    continue

      do 40 i = 2,fsize-1
         do 30 j = 2,fsize-1
            f(i,j) = buff(i,j)
30       continue
40    continue

      end
c-----------------------------------------------------------------------
c---------Histogram Equalization
c---------TJ Klausutis
c---------8/8/89
      SUBROUTINE HIST(PFPA,FSIZE)

      REAL*4 PFPA(128,128),GMAX,GMIN,DELTA,
     +       TEST1,TEST2,TEST3,TEST4, PB(254)
      INTEGER FSIZE

C  THIS ROUTINE PERFORMS A HISTOGRAM EQUAILIZATION ON A 128X128
C   ARRAY OF PIXEL VALUES.  THIS CAN BE PERFORMED BEFORE OR AFTER
C   A THRESHOLD IS PERFORMED ON THE FOCAL PLANE IMAGE
C  ADDED TO THE DSM 7-27-89 BY TJK

C  INTIALIZE THE PROBABILITY DENSITY FUNCTION
      DO 11 K=1,254
      PB(K)=0
 11     CONTINUE

C  FIND THE MIN AND MAX VALUES IN THE PFPA ARRAY OF VOLTAGES.
      GMAX=PFPA(1,1)
      GMIN=PFPA(1,1)

      DO 10 I=2,127
      DO 20 J=2,127
      IF (PFPA(I,J) .GT  GMAX) THEN
      GMAX = PFPA(I,J)
      END IF

      IF (PFPA(I,J)  LT. GMIN) THEN
      GMIN = PFPA(I,J)
      END IF

 20     CONTINUE
 10    CONTINUE

C  CALCULATE DELTA FOR 254 EVEN INTERVALS BETWEEN GMIN AND GMAX
```

```
c*** Determine slopes and plotting of slope
      stb = -slpsz
      edb = slpsz
      do 2 i = 2,fsize - 1
        do 1 j = 2,fsize - 1
          if (pfpa(i,j) .ne. 0) then
            do 4 c = 2,fsize - 1
              do 3 d = 2,fsize - 1
                if (i-c.ne.0) then
                  m = (j - d)/(i - c)
                else
                  m = fsize + 1
                endif
                  b = j - (m*i)
                  slope(int(m),int(b/sc)) = slope(int(m),
     *              int(b/sc)) + 1
                  if (slope(int(m),int(b/sc)).gt.max) then
                    max = slope(int(m),int(b/sc))
                  endif
3             continue
4           continue
          endif
1       continue
        print*,'ROW = ',i
2     continue

c-------Thresnold slope array
      print*,'Thresholding slope()'
      print*,' MAX =',max
      print*,  THRSH =',max* 75
      count = 0
      do 8 m = -fsize,(fsize+1)
        do 7 b = -slpsz,slpsz
          if (slope(m,b).lt.(max* 75)) THEN
            slope(m,b) = 0
            count = count + 1
          endif
7       continue
8     continue
      print *,'# of entries removed =  ,count
      print*,'% of entries removed = ',(count*100)/((4*fsize*slpsz)+1)

c-------Cross-reference data in slope() with pfpa()
      stb = -slpsz
      flag = 0
      do 15 b = -slpsz,slpsz,-1
        do 14 m = -fsize,fsize+1
          if (slope(m,b).ne.0) then
            stb = b
          endif
14      continue
        print* 'B(step 1) = ',b
15    continue
      edb = slpsz
      flag = 0
      do 17 b = slpsz,-slpsz,-1
        do 18 m = -fsize,fsize+1
          if (slope(m,b) .ne.0) then
            edb = b
          endif
18      continue
      print*,'B(step 2) = ',b
17    continue
      print*,'Cross-referencing data in slope() with pfpa()'
      do 11 i = 2,fsize-1
        do 9 j = 2,fsize-1
          if (pfpa(i,j).ne.0) then
            flag = 0
            do 13 m = -fsize,(fsize+1)
              do 12 b = stb,edb
                if (j.eq.((m)*i)+(b*sc)) then
                  flag = 1
                  m = 129
                  b = slpsz
                endif
12            continue
13          continue
            if (flag.eq.0) then
              pfpa(i,j) = 0
            endif
          endif
9       continue
        print* 'ROW = ',i
11    continue

      end
c-------------------------------------------------
```

II. MultiImaging (MISP)

OBJECTIVE

Before I arrived this summer, the signal processing division had the ability to view and convolute any image at any stage of an interceptor's endgame. What they did not possess, however, was a utility allowing them to process a series of images, ala the entire flight path of that same interceptor. Sensing a need (or seeking glory), then, Troy Urquhart (the other SAI apprentice) and I set out to develop a graphic simulation to handle streams of images - a SUN workstation/vt100 variable program. What we came up with, God help us, glosses the next couple of pages. Again, for any REAL computer nerds, I've included the listing - unfortunately, sparsely "COMMENTED." My philosophy is basically that if one is not smart enough to figure out MY programming, he/she doesn't deserve to look at it.

DESIGN EXPLANATION

Since most people do not have time to sit at a SUN the entire day, I configured MISP to run either from a PC terminal (NONgraphic) or from a SUN terminal (graphic). This allowed for more convenience and friendliness between users.

Basically MISP runs with a simple command structure. The two front-ends, being graphic and terminal, are immediately set, to eliminate any user indecision or crashing throughout execution. Program control then simply loads an image up, decides if it should display it, processes it, saves any tracking data in a file, and proceeds to repeat this for every image specified by the user at the initiation of the program. After this is complete, a final evaluation is displayed, providing an instant report of how all the Signal Processing performed.

For faster writing time, Troy and I divided MISP's subroutines between ourselves. I wrote the graphics and signal processing sections, while Troy handled the I/O and user interface.

TESTING

To make sure old MISP ran, Troy and I fulfilled an immediate goal of our section - to process the entire endgame scenario of an interceptor - 1493 consecutive images. This we did, and set a record did us (or is that

we; don't I sound like YODA?). Ah, well, it felt neat to be heros.

ALGORITHM EXPLANATION

GRAPHICS

The SUN carries a nice interface to graphics in its high-level function libraries, for use in its multiprocess environment SUNVIEW, or XVIEW as many will now know it. I can't really explain much here except to say "LOOK IT UP!" Graphics on this machine is just a matter of setting the proper pointers, and deciding on the right function calls (not that this didn't involve a few tricks). I believe one should examine the program listings to decide upon what I did here.

SIGNAL PROCESSING

MISP contained four major categories of user-defined processing - PreImage, clustering, centroiding, and tracking/hardbody detection. I shall endeavor here to explain each.

A PreImage algorithm is simply a program module which takes out noise, or brings out an edge, or some other helpful operation BEFORE any real multi-imaging starts. This is simply to aid any other modules or to make their jobs easier.

Clustering defines an operation to 1) decide how many different REAL targets there are, and 2) catalog those targets and differentiate between them. The routine that I wrote for MISP is really pretty simple, using an integer array to store the targets and incrementation to differentiate between them. My scanMatrix is L-shaped, and seems to perform fine under most circumstances.

Centroiding is sort of like finding a center of gravity in an object. It is, specifically, the process by which a point is located, where for certain Mathematical operations, all mass, intensity, or area can be thought of as balanced at that point. This is useful for storing locations of a target, or for a general catalog of a target's movements.

Tracking/hardbody detection describes the process that determines if the current object is indeed a missile plume (not a cloud), and if so, finds where the body of the missile is located, for guidance purposes. This is accomplished by first measuring the velocity, acceleration, and distance that the "target" has traveled in the past time interval. If these measurements fit a certain envelope, then the program calculates the

probable location of the hardbody, printing these out. Finally, the program stores this increment's data, ready for the next image.

CONCLUSIONS

Well, what did you expect? It WORKS, OK?! Ah, you probably want me to say that "Many more time-consuming modifications will have..." No way! I'm through with MISP for this summer. The apprentice part is done. Troy and I WROTE MISP in three days, and spent THREE WEEKS debugging it!!! I REFUSE to go through that again. No way.

It works. Amen (angel songs, please).

Multiple Image Signal Processor (MISP)

```
#define VERSION  . .

#include <math.h>
#include <stdio.h>
#include <strings.h>
#include <suntool/sunview.h>
#include <suntool/canvas.h>
#include <suntool/panel.h>

#define IMGEXT   ".img"
#define SGPEXT   ".sgp"
#define INTEXT   ".int"
#define AREEXT   ".are"
#define COMPRESSCMD      "compress "
#define COMPRESSEXT      ".Z"
#define UNCOMPRESSCMD    "uncompress "

#define DEFAULTSCRIPT    "/home/derek/simulation/script file"
#define DEFAULTFINAL     "/home/derek/simulation/final"
#define DEFAULTCOMPRES   1
#define DEFAULTGAINOFF   0
#define DEFAULTGAIN      1
#define DEFAULTOFFSET    0
#define DEFAULTSCENEST   0
#define DEFAULTTHRESH    1
#define DEFAULTTHRESHV   4
#define DEFAULTCLUST     1
#define DEFAULTAIMIPT    1
#define DEFAULTHARDDT    0
#define DEFAULTTRACKVC   2
#define DEFAULTAIMPT     1
#define DEFAULTTRUP      50
#define DEFAULTTRLO      1
#define DEFAULTKFOUR     1
#define DEFAULTKTHREE    1

Frame frame, panframe, sigframe;
Canvas canvas, sigwin;
Window panel;

/* MISP
 * --The Multiple Image Signal Processing simulation
 *
 * Developed July-August 1990 by HSAPs Derek Holland & Troy Urquhart
 */

#include </home/derek/simulation/Global.h>

int comores, gainoff, scenest, thresh, threshv, clust;
int aimpt, harddt, trackvc, ntar;
int trup, trlo,

int     iarr[128][128],          idarr[128][128],
        sarr[128][128],          sdarr[128][128];
int     display;

float gain, offset, icen[3][10],acen[3][10];
float dicen[3][10], dacen[3][10];
float kthree, kfour, variable;

char    filespod[50],filesgp[50],script_file[50],final_file[50],filename[50];
FILE    *scr_fl, *final_fl;

main(argc, argv)
        int     argc;
        char    **argv;
{
        extern int      display;
        char            str[3];

        while (str[0] != 'y' && str[0] != 'Y' && str[0] != 'n' &&
               str[0] != 'N')
        {
                printf("\nDo you wish to enable graphics? (y/n) ");
                gets(str);
        }
        display = (str[0] == 'y' || str[0] == 'Y') ? 1 : 0;
        if (display)
        {
                graph_set(argc, argv);
        }
        if (!display)
        {
                term_set();
        }
        exit(0);
}
final_eval()
```

```
        pw_writebackground(pw, 0, 0, 384, 384, NULL);

        for(x=0; x<384; x=x+3)
                for(y=0; y<384; y=y+3)
                        if((idarr[x/3][y/3]==1))  \
                        pw_put(pw, x+1,y+1,1);
                        \
                        if((idarr[x/3][y/3]>=2)&&(idarr[x/3][y/3]<=6))
                        pw_put(pw, x, y, 1);
                        pw_put(pw, x+1,y,1);
                        pw_put(pw, x+1,y+1,1);
                        pw_put(pw,x,y+1,1);
                        pw_put(pw,x+2,y,1);
                        pw_put(pw,x+2,y+2,1);
                        pw_put(pw,x,y+2,1);
                        }
                        if((idarr[x/3][y/3]>=7))  {
                        pw_put(pw, x, y, 1);
                        pw_put(pw, x+1,y,1);
                        pw_put(pw, x+1,y+1,1);
                        pw_put(pw,x,y+1,1);
                        pw_put(pw,x+2,y,1);
                        pw_put(pw,x+2,y+1,1);
                        pw_put(pw,x+2,y+2,1);
                        pw_put(pw,x+1,y+2,1);
                        pw_put(pw,x,y+2,1);
                        }
                }
        }
}
graph_proc()
{
        extern int      sarr[128][128], sdarr[128][128];
        extern char     filename[50];

        register Pixwin *sw;
        register int    i,j,x,y;
        u_char          red[128],
                        blue[128],
                        green[128];


        sw = canvas_pixwin(sigwin);

        j = 0;
        for(i=0;i<128;i++)  {
                j += 2;
                if(j==256)
                    j = 255;
                blue[i] = green[i] = red[i] = j;
        }

        pw_setcmsname(sw, "HSAP");
        pw_writebackground(sw,1,1,384,384,NULL);

        for(x=1;x<384;x=x+3)  {
                for(y=1;y<384;y=y+3)  {
                        if((sdarr[x/3][y/3]==1))  {
                        pw_put(sw, x+1,y+1,1);
                        }
                        if((sdarr[x/3][y/3]>=2)&&(sdarr[x/3][y/3]<=6))  {
                        pw_put(sw, x, y, 1);
                        pw_put(sw, x+1,y,1);
                        pw_put(sw, x+1,y+1,1);
                        pw_put(sw,x,y+1,1);
                        pw_put(sw,x+2,y,1);
                        pw_put(sw,x+2,y+2,1);
                        pw_put(sw,x,y+2,1);
                        }
                        if((sdarr[x/3][y/3]>=7))  {
                        pw_put(sw, x, y, 1);
                        pw_put(sw, x+1,y,1);
                        pw_put(sw, x+1,y+1,1);
                        pw_put(sw,x,y+1,1);
                        pw_put(sw,x+2,y,1);
                        pw_put(sw,x+2,y+1,1);
                        pw_put(sw,x+2,y+2,1);
                        pw_put(sw,x+1,y+2,1);
                        pw_put(sw,x,y+2,1);
                        }
                }
        }
}
static viewimgj(), quitprj(), run_simulationj();

graph_set(argc, argv)
        int argc;
        char **argv; {
```

```
                gets(inpt);
        }
        gainoff = inpt[0] - 48;

if (str[0] == 'D')
{
        printf("Scene stabilization:\n");
        printf("\t0) none\n");
        printf("\t1) Standard SPPD algorithms\n");
        while (inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        scenest = inpt[0] - 48;
}
if (str[0] == 'E')
{
        printf("Threshold\n");
        printf("\t0) none\n");
        printf("\t1) Fixed value\n");
        printf("\t2) Percentage threshold\n");
        while(inpt[0] != '0' && inpt[0] != '1' &&
                inpt[0] != '2')
        {
                gets(inpt);
        }
        thresh = inpt[0] - 48;
}
if (str[0] == 'F')
{
        printf("Clustering.\n");
        printf("\t0) none\n");
        printf("\t1) Standard SPPD algorithms\n");
        while(inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        clust = inpt[0] - 48;
}
if (str[0] == 'G')
{
        printf("Tracking\n");
        printf("\t1) By Area centroid\n");
        printf("\t2) By intensity centroid\n");
        while (inpt[0] != '1' && inpt[0] != '2')
        {
                gets(inpt);
        }
        trackvc = inpt[0] - 48;
}
if (str[0] == 'H')
{
        printf("Aimpoint offset:\n");
        printf("\t0) none\n");
        printf("\t1) Standard SPPD algorithms\n");
        while(inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        aimpt = inpt[0] - 48;
}
if (str[0] == 'I')
{
        printf("Hardbody detection:\n");
        printf("\t0) none\n");
        printf("\t1) Standard SPPD algorithms\n");
        while (inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        harddt = inpt[0] - 48;
}
if (str[0] == 'Z')
{
        printf("Uncompression/Compression:\n");
        printf("\t0) off\n");
        printf("\t1) Standard\n");
        while (inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        compres = inpt[0] - 48;
}
if (str[0] == '0')
{
        return;
}
if (str[0] == 'a')
```

```
                                printf("Gain = ");
                                mispget();
                                gain = variable;
                        }
                        if (str[0] == 'b')
                        {
                                printf("Offset = ");
                                mispget();
                                offset = variable;
                        }
                        if (str[0] == 'c')
                        {
                                printf("Threshold = ");
                                mispget();
                                thresnv = variable;
                        }
                        if (str[0] == 'd')
                        {
                                printf("Max. target movement/frame = ");
                                mispget();
                                trup = variable;
                        }
                        if (str[0] == 'e')
                        {
                                printf("Min  target movement/frame = ");
                                mispget();
                                trlo = variable;
                        }
                        if (str[0] == 'f')
                        {
                                printf("K3 constant = ");
                                mispget();
                                kthree = variable;
                        }
                        if (str[0] == 'g')
                        {
                                printf("K4 constant = ");
                                mispget();
                                kthree = variable;
                        }
                }
        }
        menu_display()
        {
                extern  char    script_file[50],
                                final_file[50];
                extern  int     gainoff, scenest, thresh,
                                thresnv, clust, aimpt, harddt, trackvc, trup,
                                trlo;
                extern  float   gain, offset, kthree, kfour;

                printf("\n\nMISP setup \n\n");
                printf("A) Script file\t\t\"%s\"\n",script_file);
                printf("B) Final file\t\t\"%s\"\n",final_file);
                printf("C) Gain & offset\t");
                if (!gainoff)
                {
                        printf("none\n");
                }
                if (gainoff == 1)
                {
                        printf("Standard SPPD algorithms\n");
                        printf("   a) Gain = %f\n   b) Offset = %f\n",gain,offset);
                }
                printf("D) Scene stabilization.\t");
                if (!scenest)
                {
                        printf("none\n");
                }
                if (scenest == 1)
                {
                        printf("Standard SPPD algorithms\n");
                }
                printf("E) Thresnold \t");
                if (!thresn)
                {
                        printf("none\n");
                }
                if (thresh == 1)
                {
                        printf("Fixed value\n");
                }
                if (thresn == 2)
                {
                        printf("Percentage thresnold\n");
                }
                if (thresh == 1 || thresn == 2)
```

```
open_script

        extern  char    script_f: e'80],        filename,50,
        extern  FILE    *scr_f
        scr_f: = fopen(script_file, "r")
        if (scr_fl == NULL)
        {
                printf("ERROR: Could not open file. %s\n",script_file);
                printf('--MISP aborted\n');
                exit(9);
        }
        read_script();
}
static quitpr()
{
        window_destroy(frame);
        printf("MISP exit ");
}
read_script()
{
        extern  char    filename[50];
        extern  FILE    *scr_fl;
        int     count=0;

        while (filename[count-1] != '\n' && count <= 50)
        {
                filename[count] = fgetc(scr_fl);
                count++;
        }
        filename[count-1] = '\0';
        count = fgetc(scr_fl);
}

static run_simulation()
{
       .extern int      display;

/* variables for file i/o, etc  */
        extern  char    filename[50];

        extern  FILE    *scr_fl, *final_fl,
        extern  char    script_file[50],
                        final_file[50];
        extern  float   icen[3][10];
        menu();
        if (script_file[0] == '\0') { strcpy(script_file,DEFAULTSCRIPT); }
        if (final_file[0] == '\0') { strcpy(final_file,DEFAULTFINAL); }
        open_script();
        init_final();
        while(filename[0] != '\0')
        {
                zero_cent();
                load_image();
                if(icen[2][0] != 2)
                {
                        if(display==1)
                        {
                                thresh_img();
                                graph_img();
                        }
                        signal_process();
                        if (display==1)
                        {
                                thresh_proc();
                                graph_proc();
/*                              graph_cent();                   */
                        }
                        save_process();
                }
                save_final();
                swap_centroids();
                read_script();
                printf("\n");
        }
        close(scr_fl);
        final_eval();
        script_file[0] = '\0';
        printf("\nMISP EXECUTE COMPLETE.\n\n");
}
save_final()
{
        extern  char    filename[50];
        extern  FILE    *final_fl;
        extern  float   icen[3][10],    acen[3][10];
        extern  int     trackvc;
        int     cen[3][10],
                x, y;
```

```
        if (compres == '
                strcpy(cmd,COMPRESSCMD);
                strcat(cmd,name);
                system(cmd);
                strcpy(cmd,COMPRESSCMD);
                strcat(cmd,filename);
                strcat(cmd,IMGEXT);
                system(cmd);
        }
}

signal_process()
{
        extern int gainoff, scenest, thresh, threshv, clust;                    extern int
        extern int trup, trlo, ntar;
        extern int iarr[128][128], sarr[128][128];
        extern float gain, offset, icen[3][10],acen[3][10];                     extern float
        extern float kthree, kfour;

        extern char filename[50];

        int target[128][128], i, n, row, col, g, h, avei, ivel, j, y, x;
        int a, b, c, d, e, f;
        float  valux, valuy, valu;
        float xca, yca, xsm, ysm, xv, yv, acm;
        float rowmoa, colmoa, rowmom, colmom, rowtoa, rowtot;
        float total, totala, maxtwo, yaim, xaim;

        printf("\nSignal Processing image %s.\n\n", filename);
        for(x=0;x<128;x++)  {
                for(y=0;y<128;y++)  {
                        sarr[x][y] = iarr[x][y];
                }
        }

/*      Gain and Offset section   */

        if(gainoff == 1)  {
                printf("Running SPPD Gain and Offset\n");
                for(x=0;x<128;x++)  {
                        for(y=0;y<128;y++)
                                sarr[x][y] = (gain*sarr[x][y]) + offset;
                }
        }

        if(gainoff == 0)  {
                /*printf("No Gain and Offset selected\n");*/
        }

/*      Scene stabilization section   */

        if(scenest == 0) {
                /*printf("No Scene Stabilization selected\n");*/
        }

/*      Threshold Section.  */

        if(thresh == 1)   {
                printf("Threshold = %d\n",threshv);
                for(x=0;x<128;x++)  {
                        for(y=0;y<128;y++)  {
                                if(sarr[x][y] < threshv)
                                        sarr[x][y] = 0;
                        }
                }

        }

        if(thresh == 0)  {
                /*printf("No threshold selected\n");*/
        }

/*      Clustering section      */

        if(clust == 1)  {
                ntar = 0;
                for(x=0;x<128;x++)  {
                        for(y=0;y<128;y++)  {
                                if(sarr[x][y] > 0)  {
                                        if(ntar == 0) {
                                                target[x][y] = 1;
                                                ntar = 1;
                                        }
                                        else if(target[x][y-1] > 0) {
                                          target[x][y] = target[x][y-1];
                                        }
                                        else if(target[x-1][y-1] > 0) {
```

```
        extern  float    icen[3][10],    oicen[3][10];
                         acen[3][10],    oacen[3][10];
        int     x, y;

        for (x=0; x<=2; x++) { for (y=0; y<=9; y++)
        {
                oicen[x][y] = icen[x][y];
                oacen[x][y] = acen[x][y];
        }}
}

term_set()
{
        extern int display;

        printf("\n\nMISP %s\n",VERSION);
        printf("Developed by DH & TU (SAI)\n");
        run_simulation();
}

thresh_img()
{
        extern  int     iarr[128][128], idarr[128][128];
        int     x, y;

        for (x=0; x<=127; x++) { for (y=0; y<=127; y++)
        {
                idarr[x][y] = (iarr[x][y] <= 128) ? iarr[x][y] : 128;
        }}
}

thresh_proc()
{
        extern  int     sarr[128][128], sdarr[128][128];
        int     x, y;

        for (x=0; x<=127; x++) { for (y=0; y<=127; y++)
        {
                sdarr[x][y] = (sarr[x][y] <= 128) ? sarr[x][y]   128;
        }}
}

static viewimg()
{
        char    byte;
        extern  char    filesgp[50],filesppd[50];
        FILE    *fp;
        int     count, cx, cy, suffix, m;
        extern int  arr [128][128], idarr[128][128];
        char suf[3];


        while(suf[0] != 'A' && suf[0] != 'B' && suf[0] != 'C')
        {
                printf("\nEnter type of image.\n");
                printf("\tA)  Initial SPPD \n");
                printf("\tB)  Signal Processed SPPD \n");
                printf("\tC)  Automatic Selection.\n");
                printf("\nVIEW>");
                gets(suf);
        }

        suffix = suf[0] - 64;

        printf("image: ");
        gets(filesppd);
        if (filesppd[0] == '\0') {
                fprintf(stderr, "ERROR  no filename typed \n");
                exit(0);
        }

        strcpy(filesgp,filesppd);
        strcat(filesppd, IMGEXT);
        strcat(filesgp  SGPEXT);

        for (m=0; m<=1; m++)
        {
        if (suffix == 1 || (suffix == 3 && m==0))
        {
                fp = fopen(filesppd, "r");
        }
        if (suffix == 2 || (suffix == 3 && m==1))
        {
                fp = fopen(filesgp, "r");
        }
        if (fp != NULL)
        {
                for (cx=0; cx<=127; cx++) { for (cy=0; cy<=127; cy++)
```

```
                        iarr[cx][cy] = 0
                        for (count=0; count<=4; count++)
                                byte = fgetc(fp);
                                if (byte>=48 && byte<=57) {
                                        iarr[cx][cy] = iarr[cx][cy] * 10 +
                                                byte - 48;
                                }
                        }
                }}
                fclose(fp);

                printf("Scaling intensity array for display . . . ");
                for (cy=0; cy<=127; cy++) { for (cx=0; cx<=127; cx++)
                {
                        idarr[cx][cy] = (iarr[cx][cy] <= 128)?iarr[cx][cy]:128;
                }}
                displmg(suffix, m);
                if (suffix == 1 || suffix == 2) { m = 2; }
        }}
}

static displmg(suffix,m)
        int suffix, m;
{
        extern int      iarr[128][128], idarr[128][128];
        extern char     filename[50];

        register Pixwin *pw;
        register int    i,j,x,y;
        u_char          red[128],
                        blue[128],
                        green[128];

        if(suffix==1)pw = canvas_pixwin(canvas);
        if(suffix==2)pw = canvas_pixwin(sigwin);
        if(suffix==3 && m==0)pw = canvas_pixwin(canvas);
        if(suffix==3 && m==1)pw = canvas_pixwin(sigwin);

        j = 0;
        for(i=0;i<128;i++)  {
                j += 2;
                if(j==256)
                   j = 255;
                blue[i] = green[i] = red[i] = j;
        }

        pw_setcmsname(pw, "HSAP");
        pw_writebackground(pw,1,1,384,384,NULL);

        for(x=1;x<384;x=x+3)  {
                for(y=1;y<384;y=y+3)  {
                        if((idarr[x/3][y/3]==1))  {
                        pw_put(pw, x+1,y+1,1);
                        }
                        if((idarr[x/3][y/3]>=2)&&(idarr[x/3][y/3]<=6))  {
                        pw_put(pw, x, y, 1);
                        pw_put(pw, x+1,y,1);
                        pw_put(pw, x+1,y+1,1);
                        pw_put(pw,x,y+1,1);
                        pw_put(pw,x+2,y,1);
                        pw_put(pw,x+2,y+2,1);
                        pw_put(pw,x,y+2,1);
                        }
                        if((idarr[x/3][y/3]>=7))  {
                        pw_put(pw, x, y, 1);
                        pw_put(pw, x+1,y,1);
                        pw_put(pw, x+1,y+1,1);
                        pw_put(pw,x,y+1,1);
                        pw_put(pw,x+2,y,1);
                        pw_put(pw,x+2,y+1,1);
                        pw_put(pw,x+2,y+2,1);
                        pw_put(pw,x+1,y+2,1);
                        pw_put(pw,x,y+2,1);
                        }
                }
        }
        graph_cent();
}

zero_cent()
{
        extern  float   icen[3][10],
                        acen[3][10];
        int     x, y;

        for (x=0; x<=2; x++) { for (y=0; y<=9; y++)
        {
```

III. C30 Simulator & Bridgeboard

OBJECTIVE

This was, in actuality, pretty fun. The C30 is a special chip designed exclusively for image processing in the interceptor. Now, a typical vehicle uses 16 C30's for its operations, but only, say 10 are active at one time. It was thought that if the remaining 6 C30's could somehow be calculating the guidance for the interceptor, the entire guidance chips and software section could be eliminated, conserving space, etc. I had to know 1) Could I fool the C30 into doing GUIDANCE operations? and 2) Was it possible to write the entire guidance system in a language syntax the C30 could understand?

THEORY

Guidance in an interceptor can be divided into three main divisions. The first, the law generation, computes acceleration commands for the thrusters. The second, called the estimator, decides where the target is, and where it is going to be after certain thrusts, etc. The final part, the autopilot, issues the actual burn-time commands the the valves in the engines.

SIMULATOR AND BRIDGEBOARD

I had two tools to help me in this - a simulator of the chip, running on a miniVAX, and a bridgeboard design version of the C30 itself, running on a PC. I used the former for initial testing, the latter for curiosity satisfaction.

PROCEDURE

Since I had no previous experience working with guidance, and because the simulator ran assembly, through a separate, special C compiler, I had to do a LITTLE research. Initially, I modified another simulation's subroutines (GEsim - this involved getting them from FORTRAN to C), and ran them under the simulator and bridgeboard. I used an extended Kalman filter for my non-linear estimator.

RESULTS

GREAT. Everything worked. I had a slight problem with memory

allocation for the simulator, but that was fixed. If external memory can be accessed, the interceptor can run ALL signal and guidance processing through one set of chips. WOW. Feels PRETTY DARN GOOD, if I don't say so myself (which I will).

"THANKS, DEREK."

REFLECTIONS

I was foolish enough to attempt ANOTHER programming job, and I now realize that I LOVE programming. However, this sort of demented Communist crap (language translation) is not suited to my personal taste. Other people should be found to do it (like ax-murderers, etc).

SO SAITH THE PROGRAMMER.

C30 guidance code

(NOTE: The following C routines are based on a FORTRAN simulation called GEsim. I take no credit for any of the FORTRAN ideas, but many modifications to the algorithms were made, allowing me to take some credit for the C code.)

```
MSWITCH =
NTHR =

/* Switch variables */

XCOMP[6] = {
YCOMP[6] = {
DCX[6] = {
DCY[6] = {
DCZ[6] = {
RATE[3] = {
POS[3] = {
AIX =
AIY =
AIZ =
AIXY =
FMASS =
XCOMP =
YCOMP =
ZCOMP =
DCX =
DCY =
DCZ =
XCG =
YCG =
ZCG =

/* ATLOG4-atlog6 VARIABLES */

ITORQ[3] = {

/* MEAS Variables */

HL[3] = {

/* LINMEAS Variables */

XK[9] = {
HB[27] =
R3 =

/* SUBROUTINE CALLS
 *
 * Find out order from somebody .
 */


}
/*   Guide   */
/* Here is the guidance laws subroutine */

guide()
{
extern double RDOT, WX, WY, WZ, UK[3], XHAT[12], GC, GAC, GV, GA
extern double TRI2B[9], RANGE, UI[3], GTERR, AT, WT, THO, TTOGO, TGO,
extern double TGOL, TEMP, SUBTR, DELXT;

extern int IGUIDE, NDOF;

    double URX = 0.
    double URY = 0;
    double URZ = 0;
    double A = 0;
    double YMISS = 0.

/*  Relative position of projectile and target */

    URX = XHAT[1]/RANGE;
    URY = XHAT[2]/RANGE;
    URZ = XHAT[3]/RANGE.

    if (IGUIDE > 1)
       goto 10;

/*  Pro-NAV Guidance Routine

    Compute Accel  commands in the inertial frame */

    UI[1] = GC*RDOT*((URY*WZ)-(URZ*WY));
    UI[2] = GC*RDOT*((URZ*WX)-(URX*WZ));
    UI[3] = GC*RDOT*((URX*WY)-(URY*WX));

/*  Augmented Guidance Routine   */

    if (IGUIDE == 1)
      {
       UI[1] = UI[1] + (GAC*XHAT[7]);
       UI[2] = UI[2] + (GAC*XHAT[8]);
```

```
        PK[i,j]=0;
      for(j=1;j<=6;++j)
        RK[i,j]=0
    *  init  the state error covariance matrix  *
      for(p=1;p<=N;++p)
        PK[1+(p-1)*(N+1)] = SIGXH[p];
   /*  init. the noise & measurement error matrices  */
      RK[1] = SIGR[1];
      RK[4] = SIGR[2];
  }
  /*  init. formal systems NLF  */
  if(IKAL == 3)  {
  /*  init   std. dev. for FS NLF  */
     for(ns1=1;ns1<=3;++ns1) {
        TOBSSD[ns1] = sqrt(SIGR[ns1]);
        GOBSSD[ns1] = sqrt(SIGRS[ns1]);
     }
  /*  init. the NLF nodes  */
     NDES = 1;
     for(JJ=1;jj<=N;++JJ)
        NDES = NDES*NPTS[JJ];
     ISTP[1] = 0;
     II = 2;
     JJ = 2;
     KK = 1;
     for(i=1;i<=N;++i)  {
        for(j=II;j<=JJ;++II)
           ISTP[j] = ISTP[j-II+1] + KK;
        II = JJ + 1;
        JJ *= 2;
        KK *= NPTS[i];
     }
  /*  init P to uniform density  */
     PNLF = 1/NDES;
  }
}

/*  Kaltime  */
/*  This is the first of the Kalman Filter Routines   Be sure
    To observe the Pointers
 */
kaltime()
{
 extern double SPHI[144], XHAT[12], B[27], UI[3], PK[144], DQ[144];
 extern double DUM2[144], SIGQ[144], DUM1[144], DTK, DLAM, DELT, MC;
 extern int N;

 int RO;

 double *PSPHI, *PXHAT, *PDUM1, *PB, *PUI, PDUM2, *PMC, *PPK, *PDQ;
 int *PN, *PRO;

 PSPHI = &SPHI;
 PXHAT = &XHAT;
 PDUM1 = &DUM1;
 PN = &N;
 PRO = &RO;
 PB = &B;
 PUI = &UI;
 PDUM2 = &DUM2;
 PMC = &MC;
 PPK = &PK;
 PDQ = &DQ;

 RO = 1;

 if(DTK != DELT)  {
       DTZ = DELT
       sysmat();
 }

 dmult(PSPHI, PXHAT, PDUM1, PN, PN, PN, PN, PN, PRO);
 dmult(PB, PUI, PDUM2, PN, PMC, PN, PN, PMC, PRO);
```

```
        index = index + *(nra);
        index = index + *(nra);

}

/*  Ddiag  */
/*  This is another one of those troublesome  d' routines
    It zeros part of a matrix
 */

ddiag(a, const, nra, n)

double *a, *const;
int *nra, *n;

{
 int index, irow, icol, i;

 index = 0;
 for(icol=1;icol<=(*(n));icol++)  {
        for(irow=1;irow<=(*(n));irow++)
                *(a+(index+irow)) = 0;
        index = index + *(nra);
 }

 index = 1;
 for(i=1;i<=(*(n));i++)  {
        *(a+(index)) = const;
        index = index + *(nra);
 }

}

/*  Dinvp  */
/*  This is ANOTHER 'd' routine
 *  it calculates the inverse of a matrix and stores it in
 *  that matrix
 */

dinvp(ar, ii, jj, ipv)

double *ar, a, b, *ipv;
int *ii, *jj;

{
 int i, n, m, j, k, l;

 for(i=0;i<=(*(ii));i++)
        *(ipv + i) = i;

 for(n=1,n<=(*(ii));n++)  {

        if((n-(*(ii)))<0)goto 2;
        if((n-(*(ii)))==0)goto 11;
        goto 2;

2:      a = *(ar + n);

        if(a<0)goto 3;
        if(a==0)goto 4;
        goto 4;

3:      a = -a;
4       i = n
        m = n + 1

        for(j=m;j<=(*(ii));j++)  {

                b = *(ar + j);

                if(b<0)goto 5;
                if(b==0)goto 6;
                goto 6;

5:              b = -b;

6.              if((a-b)<0)goto 7;
                if((a-b)==0)goto 8;
                goto 8;

7               a = b;
                i = j;
        }
8:;

        if((i-n)<0)goto 9;
        if((i-n)==0)goto11;
```

```
double e sum
int  ndexc   , irowa  irowb, indexa  indexb

for(irowb=1,irowb<=(*(n)),irowb++)   {
       for(irowa=1;irowa<=(*(l)),irowa++)   {
                indexa = irowa;
                indexb = irowb;
                sum = 0;
                for(i=1;i<=(*(m)),i++)   {
                        sum = sum + (*(a + indexa) * (*(b + indexb)));
                        indexa = indexa + *(nra);
                        indexb = indexb + *(nrb);
                }
                *(c + (indexc+irowa)) = sum;
       }
indexc = indexc + *(nrc);
}

}
/*  Dmult  */
/*  This subroutine multiplies two matrix partitions
 *  and stores the result in a third
 */

dmult(a,b,c,nra,nrb,nrc,l,m,n)

double *a, *b, *c;
int *nra, *nrb, *nrc, *l, *m, *n;

{
 double sum;
 int indexa, indexb, indexc, m, irowa, icolb;

 indexb = 0;
 indexc = 0;

 for(icolb=1;icolb<=(*(n)),icolb++)   {
       for(irowa=1;irowa<=(*(l)),irowa++)   {
                indexa = irowa;
                sum = 0;
                for(i=1;i<=(*(m));i++)   {
                        sum = sum + (*(a + indexa) * (*(b + (indexb + i))));
                        indexa = indexa + *(nra);
                }
                *(c + (indexc + irowa)) = sum;
       }
 indexb = indexb + *(nrb);
 indexc = indexc + *(nrc);
 }

}
/*   Dsubt   */
/*   This subroutine subracts two matrix partitions, storing
     the result in a third
 */

dsubt(a,b,c,nra,nrb,nrc,m,n)

double *a, *b, *c;
int *nra, *nrb, *nrc, *m, *n;

{
 int indexa, indexb, indexc, icol, irow;

 indexa = 0;
 indexb = 0;
 indexc = 0;

 for(icol=1;icol<=(*(n));icol++)   {
       for(irow=1;irow<=(*(m));irow++)   {
                *(c + (indexc+irow)) = *(a+(indexa+irow)) -
                        *(b+(indexb + irow));
       }
       indexa = indexa + *(nra);
       indexb = indexb + *(nrb);
       indexc = indexc + *(nrc);
 }

}
/*   Atlcg4    */
/* Subroutine applies actuator command logic for a four nozzle
   attitude control system */

atlcg4()
{
       external double TORQ[3], ATTFOR[6], THRNOZ[6], ITORQ[3];
```

```
*   Atlog6   *
/* Subroutine applies actuator command logic for a six nozzle
   attitude control system */

atlog6()
{
        external double TORQ[3], ATTFOR[6], THRNOZ[6], ITORQ[3];
        int             i, temp;

/* Initialize the thruster commands to zero */
        for (i=1; i<=6; i++) {
                THRNOZ[i] = 0.0;
        }

/* Convert torque commands to integer commands */
        for (i=1; i<=3; i++) {
                temp = TORQ[i];
                ITORQ[i] = temp;
        }

/* Attitude control thruster logic chart */

/* Roll torque control */
        if (ITORQ[1] == 1) {
                THRNOZ[6] = ATTFOR[6];
        }
        if (ITORQ[1] == -1) {
                THRNOZ[5] = ATTFOR[5];
        }

/* Pitch torque control */
        if (ITORQ[2] == 1) {
                THRNOZ[1] = ATTFOR[1];
        }
        if (ITORQ[2] == -1) {
                THRNOZ[3] = ATTFOR[3];
        }

/* Yaw torque control */
        if (ITORQ[3] == 1) {
                THRNOZ[2] = ATTFOR[2];
        }
        if (ITORQ[3] == -1) {
                THRNOZ[4] = ATTFOR[4];
        }
}
/*  Attcon  */
/* Attitude control routine */

attcon()
{
        external double SWTHLD[3], ATTFOR[6], RATE[3], POS[3], ANGERR[3],
                        RATERR[3], THRNOZ[6], TORQ[3];
        external double MSWITCH,NTHR;
        int             i;

/* Use switching line logic to determine the required torques */
        switch();

/* Use attitude control thruster logic to determine which
   thrusters to fire.
        NTHR = 4                four nozzle configuration
        NTHR = 6                six  nozzle configuration
   (If linear attitude control thrusters are being used, do not
    use attitude control logic chart) */
        if (MSWITCH == 1) {
                for (i=1; i<=6; i++)
                   THRNOZ[i] = 0.0;
                goto 1;
        }

/* Six nozzle attitude control logic */
        if (NTHR == 6) {
                atlog6();
                goto 1;
        }

/* Four nozzle attitude control logic */
        atlog4();
1 ,
,

/*  Switch  */

/* Switching line logic routine
```

```
            DDQ[ ][i-3]  = PHI33*SIGQ[i-3];

         for (i=7; i<=9; i++)
            DPHI[i][j] = PHI;
            DDQ[i][j-6] = DDQ[i-6][j];
            DDQ[i][j-3] = DDQ[i-3][j];
            DDQ[i][j] = PHI33*SIGQ[i-6];
         }

/* Store all matrices as column vectors */
         NR = 0
         for (j=1; j<=9; j++) {
            for (i=1; i<=9; i++) {
               PHI[i+NR] = DPHI[i][j];
               DQ[i+NR] = DDQ[i][j];
            }
            NR = NR + 9;
         }
         NR = 0
         for (j=1; j<=3; j++) {
            for (i=1; i<=9; i++) {
               B[i+NR] = DB[i][j];
            }
            NR = NR + 9
         }
}
/* linmeas */
/* This subroutine computes the linearized measurement matrix
   for the system with inertial measurements. */

linmeas()
{
         extern double XK[9],HB[27],
         extern double RANGE,R3

         int i;

         /* Zero the linearized measurement matrix */
         for (i=1; i<=27; i++)
               HB[i] = 0.0;

         /* Compute the linearized measurement matrix */
         RANGE = sqrt(XK[1]*KX[1] + XK[2]*XK[2] + XK[3]*XK[3]);
         R3 = RANGE*RANGE*RANGE;
         HB[1] = XK[1]*XK[3]/R3;
         HB[2] = -XK[1]*XK[2]/R3;
         HB[3] = XK[2]*XK[3]/R3;
         HB[4] = (XK[1]*XK[1] - XK[3]*XK[3])/R3;
         HB[5] = -(XK[1]*XK[1] + XK[2]*XK[2])/R3;
         HB[6] = -XK[2]*XK[3]/R3;
}


/*  Meas
 *
 *     Something about computing measurement for
 *  the system ----->     Z(k) = h(x(k)) (inertial)
 */

meas()
{
 extern double XK[9], HL[3], RANGE;

 RANGE = sqrt(XK[1]*XK[1]+XK[2]*XK[2]+XK[3]*XK[3])
 HL[1] = -XK[3]/RANGE
 HL[2] = +XK[2]/RANGE
}
```

IV. Miscellaneous

Here I decided to put any other crapola (a term not original; first coined by my Music teacher) that didn't fit in with everything else (Basically, the projects in which I ASSISTED).

A. Hough Transform (Troy Urquhart)

Basically the Hough takes any image and produces a crescent of the outside pixels. I supplied the original program, but TROY took this scanty thing and made it into a masterpiece.

B. Neural Nets

Didn't do much here. Just assisted in initial design.

C. Other Crapola (same term)

Basically other diagrams or pictures.

```
RDOT =
WX =
WY =
WZ =
   [3] = {
.HAT[12] = {
GC =
GAV =
GV =
GA =
TRI2B[9] = {
RANGE =
UI[3] = {
GTERR =
WT =
AT =
THO =
TTOGO =
TGO =
TGOL =
DELXT =
IGUIDE =
NDOF =

/* sysmat variables */

DPHI[9][9] = {
DB [9][3] = {
DDQ[9][9] = {
SIGQ[3] = {
PHI[81] = {
   27] =
DQ[81] = {
P13 =
P23 =
P33 =
PH113 =
PH123 =
PH133 =
PH223 =
PH233 =
PH333 =
DT =
DLAM =
TEMP =
UR =

/* Attcon Variables */

SWTHLD[3] = {
ATTFOR[6] = {
RATE[3] = {
POS[3] = {
ANGERR[3] = {
RATERR[3] = {
THRNOZ[6] = {
TORQ[3] = {
  SWTICH =
  AR =
```



Conclusion

And so, in summary, I would like to say...

NOTHING.

Well, sort of:

Thanks to you, the reader, for suffering through this comedy of errors - this mockery of English - my report. Don't worry, I won't write many more like this, and even if I do, you will not read them.

Lots of things were learned this year. I have neither the space, nor the patience to relate them all, so just trust me. I mean, come on! Does anyone actually believe I knew all this STUFF when I arrived?

Finally, all my projects were successful this year, bringing my life total to 3 and 100034 - just kidding, it's worse...

MSIS:

Multi-Sensor Integration System

Christine D. Riendeau
AFATL / AGS
Mr. John Walker, mentor
August 15, 1990

I. Acknowledgements

I would like to extend my deepest appreciation to the following people for their support, guidance, and especially their knowledge.

Mr. Johnny Walker, mentor

Mr. John Wolverton

Major Paul Vergez

Mr. Al DiSalvio

Mr. Darryl Huddleston

Mr. James Norton

Mr. Dave Rogers

2Lt. Eric D'Anna

Mr. Reid Harrison

Finally, I would like to thank the two sponsors of our program, Mr. Don Harrison and Dr. Klausutis, for the wonderful opportunity to learn and work at the Eglin Armament Laboratory.

II. Introduction

This paper is an introduction to the Multi-Sensor Integration System. The paper is divided into several parts. The first part is taken from basic research done at the beginning of the summer; first an overview of the MSIS system and next a closer look at the millimeter wave radar subsystem. The second part of the paper describes actual experiments and tests done with the radar subsystem over the summer. Finally, the paper explains the future goals of the MSIS system and summarizes its place in the Eglin laboratory.

III. Description of Research

The MSIS System

The Multi-Sensor Integration System (MSIS) is a project designed to permit simultaneous collection of data from several types of sensors. The project will integrate three separate subsystems to permit in-house capability for the collection and analysis of target and background data. The three subsystems are made of 1) infrared (IR) and TV cameras, 2) an X-band search and track radar, and 3) a millimeter wave (MMW) radar. Because this combination of sensors permits tracking both air and ground targets, the tower facility on the roof of Building 13 was selected for the site of the MSIS system. Some equipment for the MSIS system (i.e., the MMW radar and the IR cameras) is still being procured. The infrared and TV cameras (EO/IR) provide additional background information which will be useful in distinguishing live targets from dead ones, and which will be extremely effective when developing software for the dual mode sensors (see CONCLUSIONS). The EO cameras provide both wide-angle and boresight video capabilities. All cameras are mounted upon the track mount (still in procurement) that will also be used to point the MSIS sensors under computer control.

The X-band search and track radar is more familiarly known as the DIVAD. It is designed to track airborne targets flying at medium range from the tower (about 10 km); because the physical antenna limitations restrict the tracking antenna to positive elevation angles, tracking ground targets on the DIVAD will not be possible. The X-band radar has internal in-phase and quadrature analog radar video signal lines;

because the DIVAD is a Doppler radar system, it can determine target velocity and distinguish between helicopters and fixed-wing targets.

Finally, the MSIS system includes a millimeter wave radar that operates at a frequency of 94 GHz. It is called the MMW radar because its wavelength is 3 millimeters. The purpose of this radar is to locate ground targets, both stationary and moving, and there are several different types of radar sensors that could be used and often interchanged (figure B).

The Millimeter Wave Radar

The millimeter wave radar that has been selected for use in the MSIS system is currently being upgraded. An additional new single-channel receiver is designed to be capable of switching from FMCW to pulse mode of operation, and both the transmitter and receiver have separate antennas. This is called a bistatic configuration. It is formally known as the I-94 radar, as it operates at a frequency of 94 GHz. The I-94, to have target tracking capability, must be attached to the track mount with the EO/IR cameras. Originally, the I-94 was designed as a monopulse radar with eight radar video signal outputs. In a monopulse radar the transmitter and receiver share a single aperture (monostatic), which is made of four individual antennae that allow the radar to track moving targets. A single aperture for transmitting and receiving is defined as monostatic. The first four video signal outputs (channels) are electronic reconstructions of the beams from the antennae. The receiver has four more channels rather than one, which are labeled the sum, the azimuth difference, the elevation difference, and the azimuth-elevation difference channels. These channels tell the distance and direction

to the target. The transmitter used in both radars is the same, and the monopulse and I-94 radars can both be used in the MSIS system. Because the I-94 and monopulse radars are not yet available to work with in the laboratory, the hands-on experience was gained through using the simplest millimeter wave radar. It is called the beacon, but is actually more than its name implies. The beacon is a one channel, one antenna, 94 GHz radar that can generate its own timing or use timing supplied from a remote source. Because the beacon had not been used for a very long time, the first task was to characterize it and make sure all parts were working.

IV. Description of Projects

Testing the Radar IF System

A radar's IF system begins once the input from the antenna is mixed with the output of the local oscillator (figure C1). The resulting frequency is called the Intermediate Frequency (IF). The IF output goes through several devices before being turned into the video output from which targets are detected (figure C2). The radar's IF system is designed to cut out a certain block of frequencies by using high-pass and low-pass filters. The beacon also provides a test point where the IF output can be tested before it is converted to video signal.

To characterize the beacon, each piece or group of equipment in the IF system has to be tested, both to see whether it is working properly and to pinpoint exact characteristics. The test set-up (figure D) provides RF power from the sweep generator, through attenuators, to the device under test. A detector takes the output from the device and converts it to video signal and transfers it to the scalar analyzer, which plots the output on the screen. Should the device under testing be an attenuator, the results on the scalar analyzer should show a decrease in power from the sweep generator. When testing an amplifier, results should show an increase in power, and the increase should be linear. During the test of the amplifier in the beacon's IF system, the results were found to show an increase of 35 dB when the power from -77 to -31 dBm was applied to the input of the amplifier (figure E). This plot is called the transfer function of the amplifier. The graph of this data shows that the amplifier gain is linear over an input range of -77 to -31 dBm.

Timing Signals

One major step in developing the MSIS system is to develop software that permitsf timing control of the I-94 radar through the LeCroy boxes and the Compaq computer. Because the timing for the beacon could be supplied externally as well as internally, it gave a head start when developing the computer programs that would work with a similar radar. Timing for the I-94 will differ from timing for the beacon because the I-94 has two antennae, one for the transmitter and another for the receiver. This eliminates the need for several steps in timing, such as the receiver protection switch.

Tracking Targets

When tracking targets with the beacon, the radar has to be guided by hand. It is mounted on a tripod with wheels that change azimuth and elevation. The I-94 radar's tracking will be controlled through the LeCroys in conjunction with the track mount. Defining targets that are being detected was often a guess, because targets of opportunity were used.

The need for greater beamwidth in tracking targets was felt, and after the IF system had been characterized and found without great fault, the next step was to change the antenna to give a greater beamwidth. The original antenna had a twelve-inch aperture, which gave a beamwidth of about one-half a degree. This was calculated using a computer program created by David Rogers to calculate beamwidth from a known wavelength and circular aperture.

The antenna was removed and replaced with an antenna whose aperture measured 6 inches, and the beamwidth was doubled to approximately one degree (figure F).

Finally, the area covered by the beam at a distance of 400 meters was calculated for each antenna using standard trigometric functions (figure G). These functions determine the radius of the circular area covered by the beam, but this area is circular only if the beam is perpendicular to the target plane. Supposing that the target is a flat plane, at four hundred feet a beamwidth of .5 degrees could cover an area with a diameter of 1.21 meters, while a beamwidth of 1 degree would cover an area with a diameter of 2.24 meters.

V. Conclusions

Future Goals

One goal for the MSIS system that will be realized in the near future is control of the system through the LeCroy boxes and the Compaq computer. Because of the absence of the I-94 radar, it was impossible to complete the set-up at this time, but using the beacon allowed a start on software development. Working with the beacon also helped to eliminate some interface problems between the Compaq, LeCroys, and the millimeter wave radar.

Another goal, this one long-term, of the MSIS system developers is to develop the algorithms that will be used to combine information from the different types of sensors. Although now each sensor has to be operated independently from the others, the plan is that in the future these algorithms will enhance the development and operation of dual mode sensors.

Summary

The Multi-Sensor Integration System is made of three separate modes of sensors--the infrared and TV cameras, the DIVAD track and search radar and the 94 GHz millimeter wave radar called the I-94. The I-94 radar is in procurement; in its absence a single-channel radar called the beacon was calibrated and set up to help eliminate interface problems and to allow a start on software development.

The first objective this summer was to activate the beacon, then to calibrate and characterize the radar. Once the beacon was deemed operational, several steps were

taken to improve target tracking sensitivity. The first was a testing of the IF system, the parts of a radar after input from antenna has been combined with the local oscillator output and before the detector. Graphs and plots show that all parts of the IF system, as well as the whole system itself, were working as they were supposed to. The second step involved changing antennas to give a larger beamwidth. The present antenna had a twelve-inch antenna and a beamwidth of one-half a degree. The antenna size was halved, and the beamwidth was doubled.

Finally, the beacon was controlled through its remote operation mode by the LeCroy boxes and the Compaq computer. Although it is not exactly like the I-94 radar that is being developed, interfacing the beacon with the LeCroy and Compaq eliminated some basic problems. Software to collect and combine data from all different types of sensors is now being developed.

Future goals of this system are control of the millimeter wave radar through the Compaq and LeCroys and the development of algorithms that will combine the information from two different types of sensors to obtain a clearer picture of the target. These algorithms will be used with future dual mode sensors.

VI. Reference

Graf, Rudolf F. Modern Dictionary of Electronics. Indianapolis: Howard W. Sams & Co., 1962.

Johnston, Stephen L. Millimeter Wave Radar. Dedham, MA: Artech House, 1980.

Kraus, John D. Antennas. New York: McGraw-Hill, 1988.

Stimson, George W. Introduction to Airborne Radar. El Segundo, CA: Hughes Aircraft Company, 1983.

Figure A: Multi-Sensor Integration System

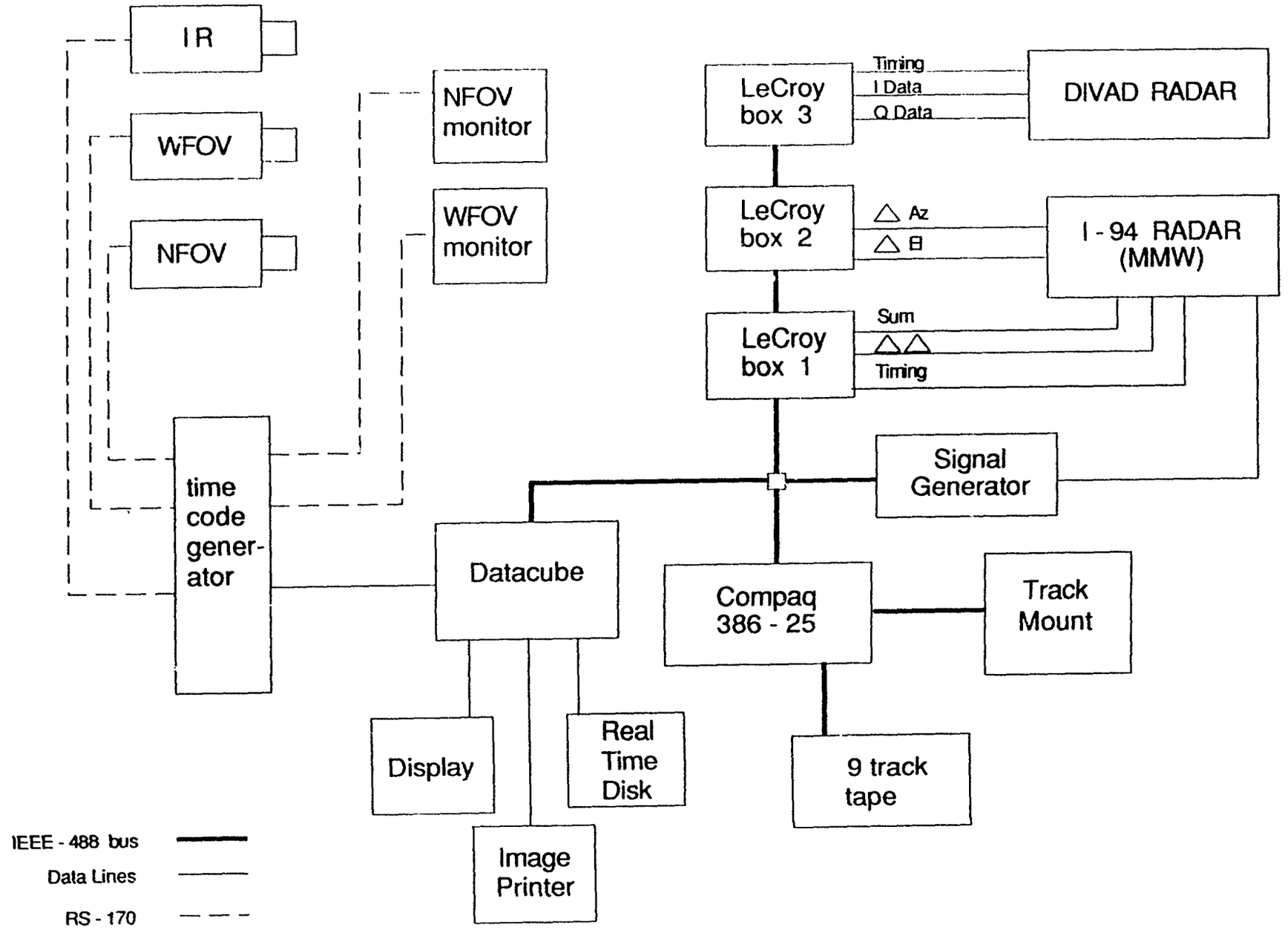


Figure B: Different Configurations of the MMW RADAR

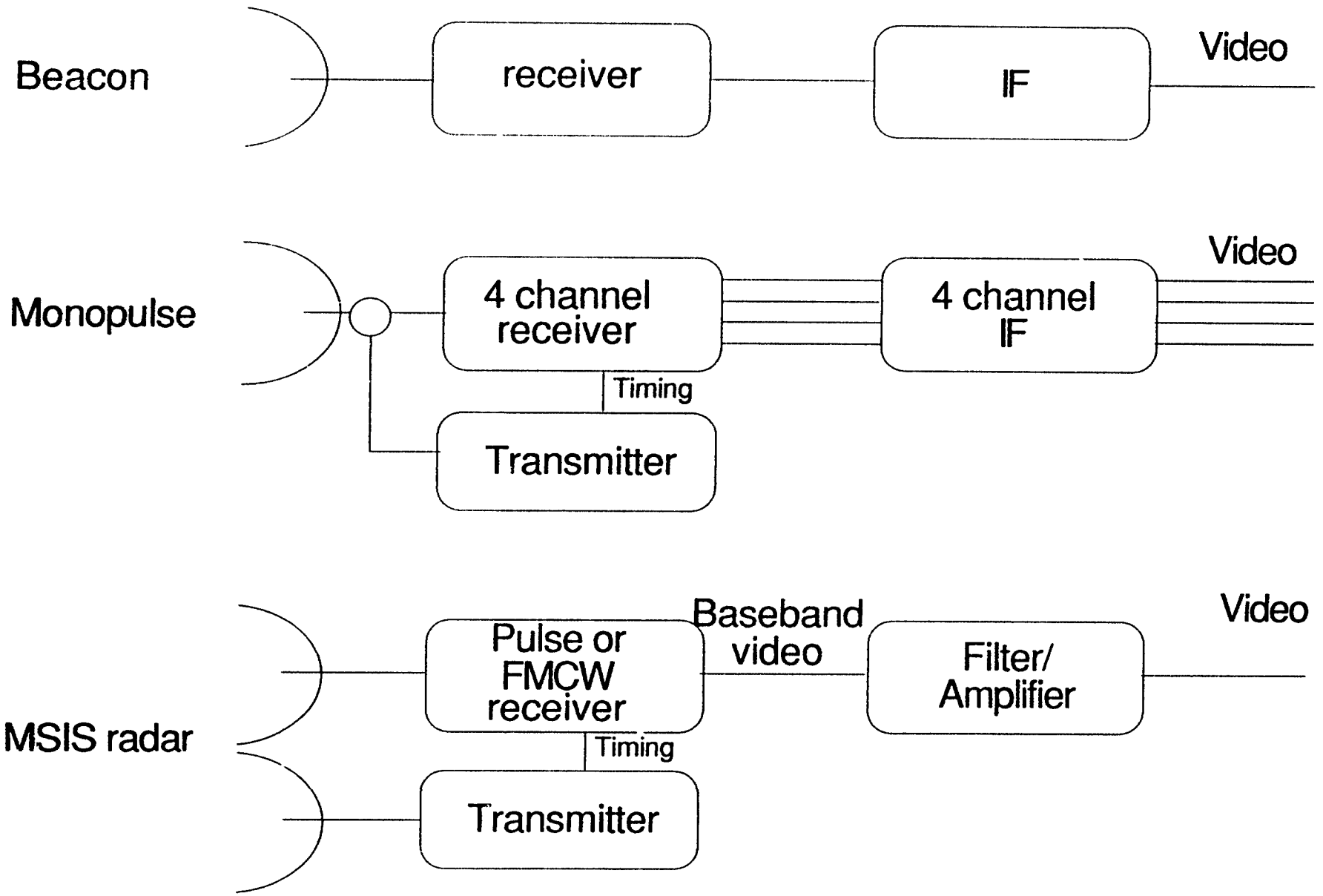


Figure C1: Diagram of Beacon

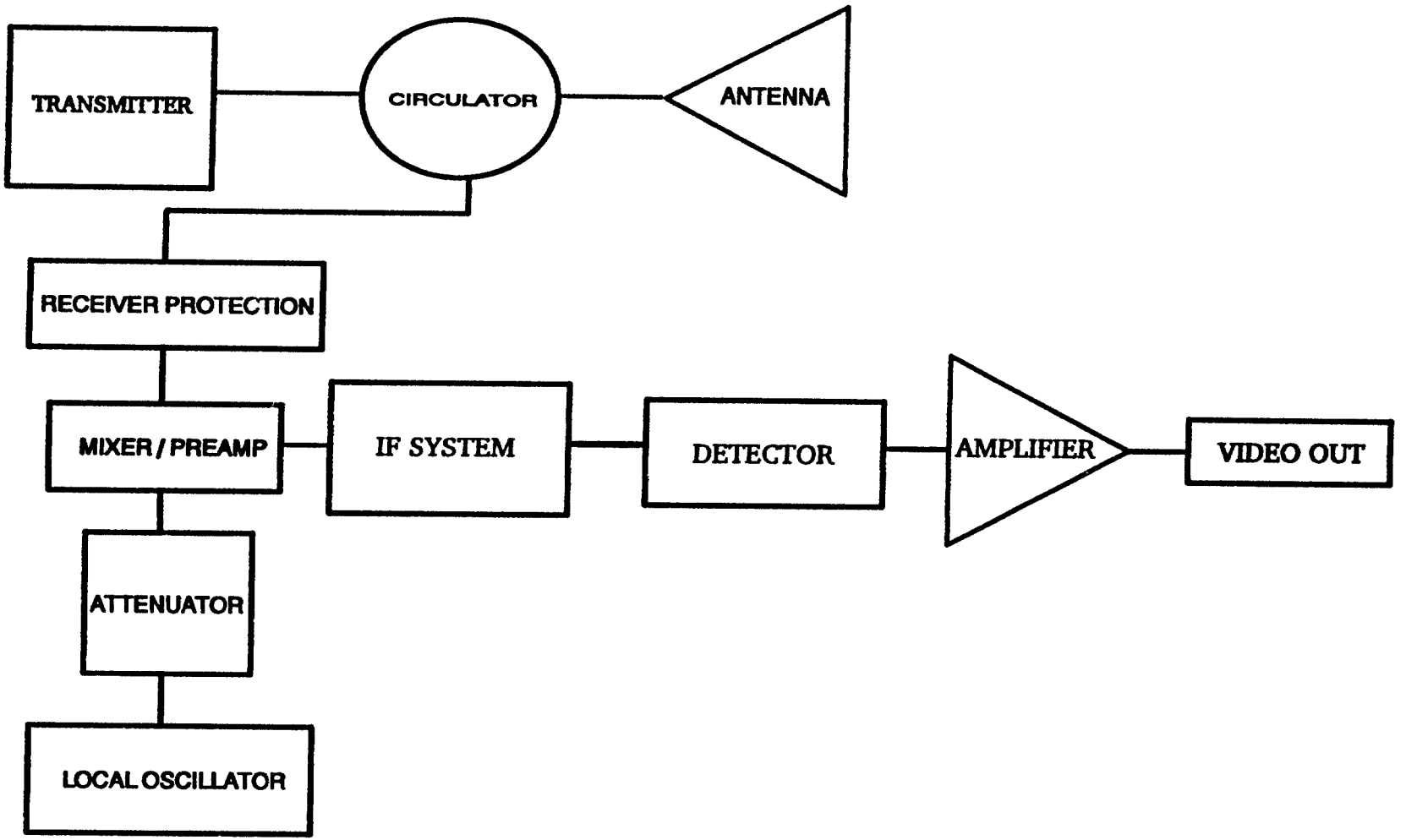


Figure C2: Diagram of IF System

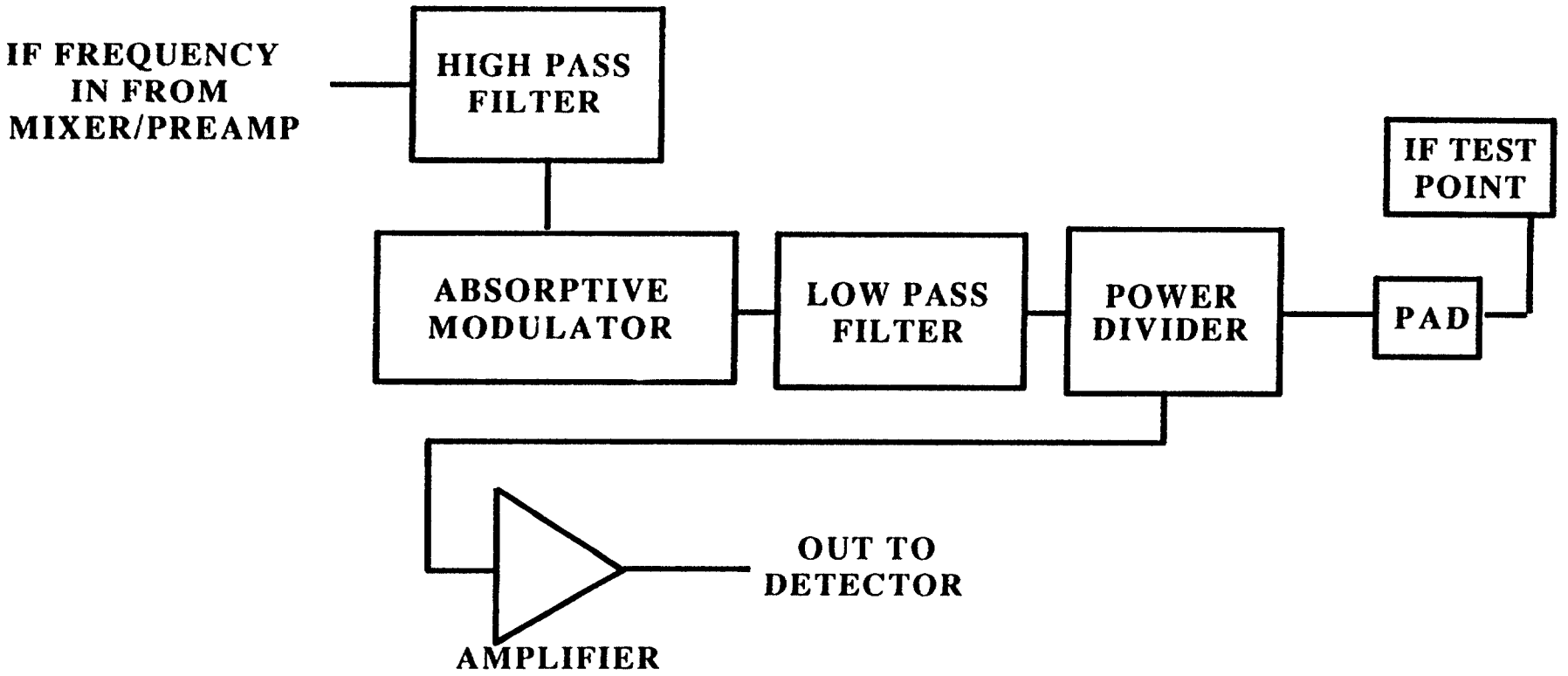


Figure D: Device Testing Set-up

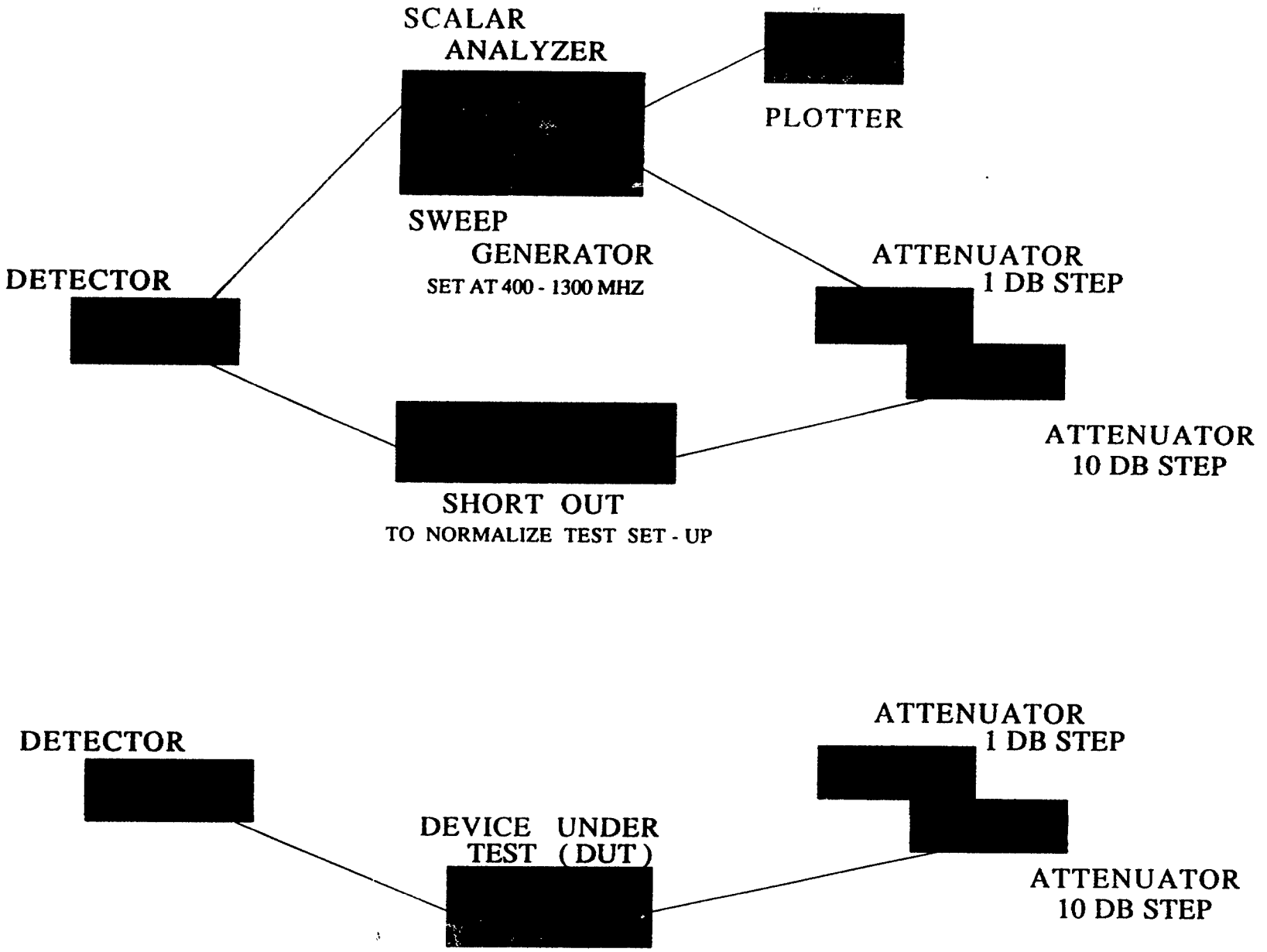


Figure E: Transfer Function of the Amplifier

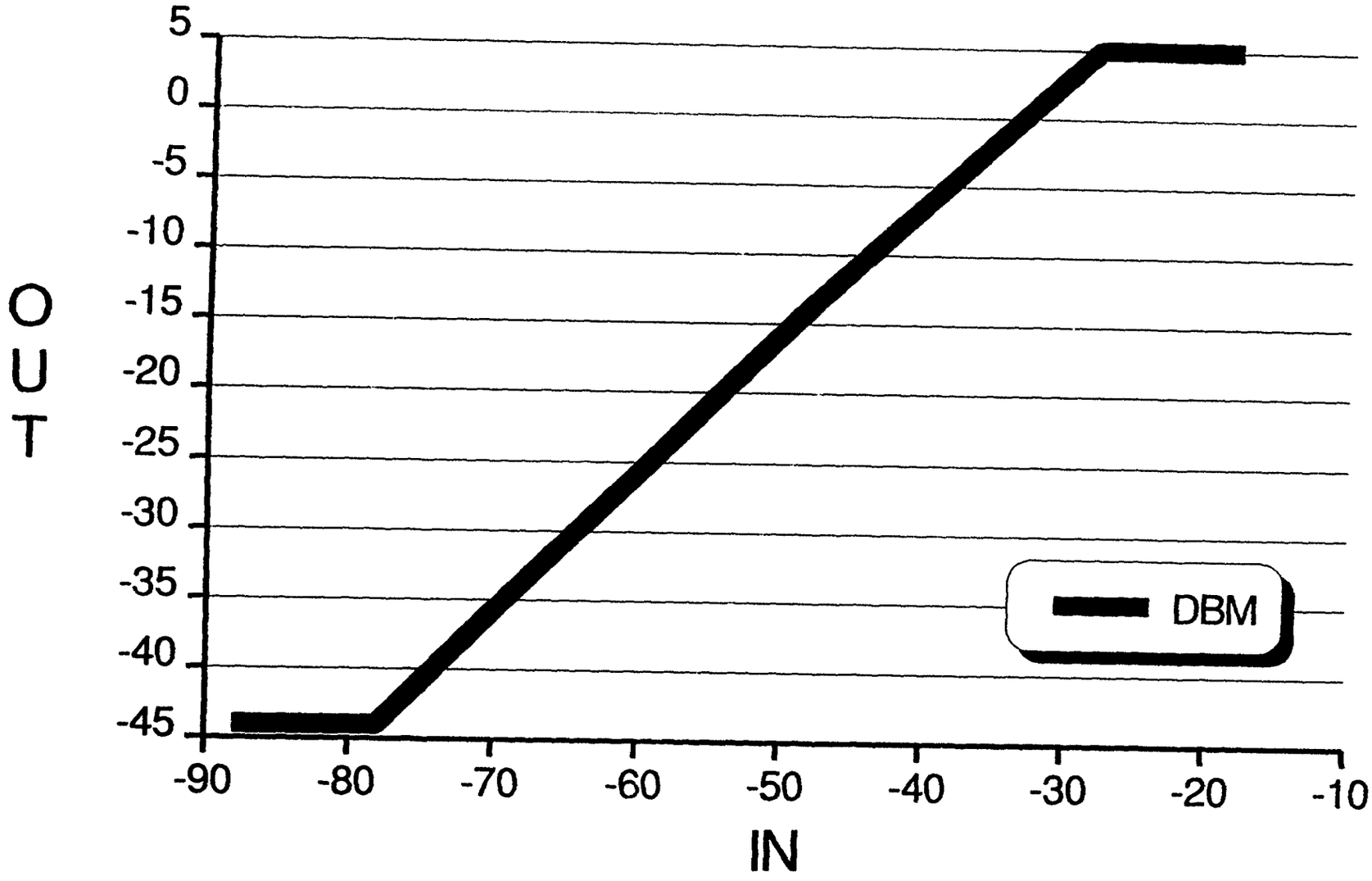
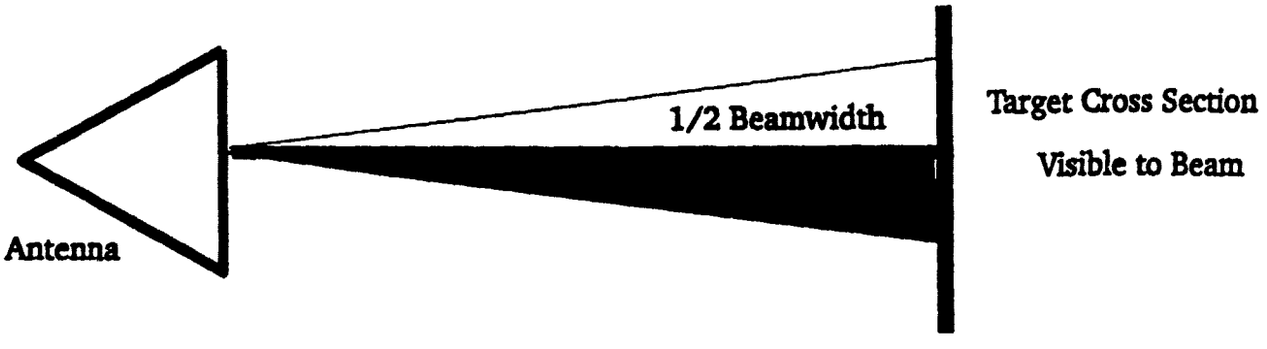


Figure F: Formula for Converting Aperture to Beamwidth

$$\theta = \frac{(51)\,\text{Frequency}}{\text{Diameter}} \qquad 1$$

Figure G: Formula for Cross Section of Target

$$\text{Tan}\,(1/2\ \text{Beamwidth}) = \frac{\text{Cross Section}}{\text{Diameter}} \qquad 2$$



ENHANCEMENT OF

RTD 710A

INTERFACE

LISA A. SCHMIDT

CAPTAIN PAUL SCHOMBER (MENTOR)

AFATL/MNE

HSAP SUMMER 1990

ACKNOWLEDGEMENTS

I would like to thank the following people for all the help and friendship they have given me. They have taught me a lot about technology and myself, and have made my past two summers at the HERD memorable ones:

Capt. Paul Schomber(mentor)
Mr. Dave Aplin
Msgt. Bobby Cole
Mr. Stacey Carswell
Mr. Rick Brumback

INTRODUCTION

This was my second summer as a high school apprentice in the Energetic Material Branch of the Air Force Armament Laboratory(AFATL) at Eglin Air Force Base. My mentor was Capt. Schomber, an officer in the U.S. Air Force and a physical chemist. He works in the High Explosives Research and Development Complex(HERD). One of the HERD's goals is to produce an insensitive high explosive. This means they want an explosive to react when it is detonated, and only when it is detonated. They do not want a bomb detonating if an adjacent bomb accidentally detonates, if there is a fire in the storage area, or if it is accidentally dropped. They want it to detonate only under specific prescribed conditions. I was involved in the dynamic testing of the explosives. Last year I helped to set up a program which allowed a computer workstation and electronic digitizer to communicate. This year my assignment was to enhance this program and expand its capabilities.

DESCRIPTION OF THE DYNAMICS LABORATORY

This summer I worked in the Dynamic Laboratory. Small scale dynamic tests of explosives are conducted there. The purpose of many of the tests is to find the detonation velocity of the explosive. The equation for velocity is distance divided by time. In order to find the distance, piezo electric pins are set up so they touch the explosive and are placed at a constant distance apart from each other. This distance is measured to the nearest .1mm and, hence, one part of the formula is fulfilled. To measure the time it takes the detonation shockwave to reach each pin, two Tektronix RTD 710A Digitizers are used. The piezo electric pins are stimulated by the detonation shockwave and transmit a signal to the digitizers. The digitizers are triggered by the piezo electric pins and record data at a rate of 5 nanoseconds per point. The digitizers then display a picture of the resulting waveform on a small screen and generate a digital readout of the data. The tests that use this procedure are:

CRITICAL DIAMETER CONE TEST
CONFINED DENT/RATE TEST
UNCONFINED DENT/RATE TEST

In the past, the process of collecting the data was very time consuming in that each and every point had to be written down by hand. Last summer the digitizers were interfaced with the Tektronix Workstation; thus allowing the two instruments to communicate. The digitizers could now be programmed by commands given through the keyboard and could send information about the recorded waveform to the computer monitor. Data reduction is still

time consuming in that this information must be transferred to another computer for analysis and plots. Enhancing last years program would allow the analysis and plot to be accomplished on the Tektronix computer, making the test quicker and more productive.

DESCRIPTION OF RESEARCH

My objective for the summer was to expand the capabilities of a program called "gpibtst" and make it easier for a person unfamiliar with the RTD 710A Digitizers and Tektronix Workstation to use. The program, developed last summer, is a C language program that enables the user to send set and query commands to the digitizers and perform a poll routine upon the instruments. When the program is implemented, a prompt is displayed informing the user that they are running this particular program. Furthermore, because there are two digitizers in the Dynamic Lab, the program allows the user to discriminate between the two instruments by typing "s" or "S" before each command to talk to scope1, and by typing any other letter in front of each command to talk to scope2. This routine is essential to the program, but a user unfamiliar with the program would not know how to use it. The prompt currectly informs the user that he has entered a program but does not tell him what the program is capable of doing. I added some information statements so as the program is entered, a message stating "You are now able to communicate with the digitizers. Please see the RTD manual for a list of commands." is displayed. I did this using a C language " for" loop and "printf" statement. I defined an integer "t" and set it equal to 0. I than said "t" could not be equal to one and incremented "t" by one. This made the program display the message in the printf statement once(when "t" is equal to zero) and than stop displaying the message. It stops because "t" increases by one and since I have instructed the computer that

"t" cannot ever equal one, the loop is terminated. This prevents the message from being displayed continuously across the screen. The user now understands what he can do, but is still unaware of how to discrimirate between scopei and scope2. Using the same loop and adding additional printf statements, the monitor displays statements instructing the user how to talk to each scope. This is how the entire loop appears in the program:

```
for(t=0;t!=1;++t){
        printf("You are now able to communicate with the\n");
        printf("Please see RTD manual for a list of GPIB\n");
        printf("commands.\n");
        printf("Type s or S in front of each command to talk\n");
        printf("to scope1, and any other letter to talk to\n");
        printf("scope2.\n");
```

There are instances when the computer is unable to communicate with the digitizer, or the computer cannot read the scope1 or scope2 file. This can be the result of several things:1)The scopes might be turned off; 2)A GPIB cable might be lose; or 3)The user forgets to type a letter before a command to distinguish which scope he is talking to. When this happens the program displays "Response = -1." With certain C compilers the read subroutine sends a code of " -1" when it is unable to read a file. An inexperienced user would not be aware of this code. Hence, I converted the read statement into an "if" statement to tell the user what was happening. I changed the program so that if the read was equal to " -1", the computer will print a message saying "Cannot read file. Please check to see if scopes are on or if all GPIB cables are secured." This is how it appears in the program:

```
if(bytes = read(scope1,r,BUFLEN) == -1){
```

```
printf("Cannot read file. Please check to see if\n");
printf("scopes are on, or if all GPIB cables\n");
printf("are secured.\n");
```

Although this program reduces the amount of time it takes to setup the tests, the scientists in the lab still have to transfer the information from the test to plot and analyze it on another computer. This program also does not make a permanent record of the test(for example, it does not save the information to a disk file, or does not make a hard copy). Once the program is exited, all the information is lost. In order to correct this, I had the program create a disk file. I did this by using the "creat" subroutine. I named the file to be opened "store". I set "store" equal to " creat" and set up the path for "store" so it would create the file in my home directory. I set the mode for the file to read/write so a user can read the file and write corrections, etc., to the file. The subroutine appears in the program as follows:

```
store =creat("/usr/schmidt/store", O_RDWR, 1);
```

In the original gpibtst program the computer would write a GPIB command to the scopes and than read the scopes response, which was put into character array "r" and set to length "BUFLEN". I changed the program so that if the user typed in the letter "y" that it would write the character array "r" to the store file. This is what it looks like:

```
else if(s[0] =='y'){
write(store,r,BUFLEN);
}
```

Hence, all the information a user has asked for concerning the test is stored into a disk file "store". Thus, there is a

permanent record of the test which will stay on file even when the program is exited. In the original gpibtst program, if-else statements were used to distinguish between communications to scope1 and scope2. If the first letter typed was a "s" or "S" the program wrote to scope1, else(meaning any other letter) the program wrote to scope2. Because I had to find some means of writing to the store file, however, I changed the program so there is an if statement and three else-if statements. I kept the statement with the "s" or "S" in it, but to write to scope2 the first letter of a command must be "t". In order for the information asked for to be sent to the store file the letter "y" must be typed, and to exit the program the letter "x" must be typed. This is how it appears in the program:

```
        if(s[0] =='s' || d[0] =='S'){
        write(scope1, &(s[1]), strlen(s) -1);
        if(bytes = read(scope1,r,BUFLEN) ==-1){
        printf("Cannot  read  file. Please check to see if\n");
        printf("scopes are on, or if all GPIB cables are secured.\n");
        }
        r[bytes] = '*';
        printf("Response = %d %s\n",bytes,r);
        }
        else if(s[0] =='t'){
        write(scope2, &(s[1]), strlen(s) -1);
        if(bytes = read(scope2,r,BUFLEN) ==-1){
        printf("Cannot read file. Please check to see if\n");
        printf("scopes are on, or if all GPIB cables are secured.\n");
        }
        r[bytes] = '*';
        printf("Response = %d %s\n",bytes,r);
        }
        else if (s[0] =='y'){
        write(store,r,BUFLEN);
        }
        else if (s[0] == 'x'){
        exit(0);
        }
        }
}
```

I added these instructions to the statement which appears in

the beginning of the program when the program is first initialized.

After saving the information onto a disk file, a Utek command can be used to make a hard copy of the test. Therefore, if the file should be accidentally erased, there is always a back-up record of the test. The command is called "cpio". This command copies files to input and output. To copy the information of a disk file to a diskette the command "cpio -o>/dev/rdf" is typed in. Than the user types in the name of the disk file. To find out what disk files are found on the diskette "cpio -i</dev/rdf is typed in. The monitor responds with a list of files and the number of blocks of memory each file has on that disk. Thus a disk copy of the entire test now exists.

After the storage file was created, a means of plotting points from that file was found. In order to get these points the GPIB command PLOT must be sent to a scope. The digitizer sends a buffer of approximately 1200 points to the screen. By typing "y" these points are than sent to the storage file so there is a disk record of the graph of the waveform. We discovered, however, that within the buffer were control characters which could not be read by the program. A time consuming method of recording these points and transferring them into another data file has to be used. The coordinates are typed in columns of two(x and y columns) and then can be read by the program. We also discovered the Tektronix computer uses Plot 10 STI(Standard Tektronix Interface) for its graphic applications. STI is a set of over 500 subroutines which are provided in both the FORTRAN and C languages. In this case, the FORTRAN language was used. The

program developed is called "wave". "Wave" consists of four major subroutines. They are tinit, array, axes, and graph. The first subroutine called is tinit which initializes the terminal parameters. The next subroutine is array. Array opens a data file and loads the data array for x and y values. Because all tests are different and give different coordinates, the program calculates the maximum value of both x and y so the plot can be drawn to scale in accordance with a particular test. The next subroutine, axes, draws the vertical and horizontal axes. It also puts the tic marks on the axes. It draws the axes by moving to a certain point and than drawing to each successive point. This is also the case in the subroutine graph, which takes the data from the arrays and plots the waveform. Each point has 400 added to it because when drawing the axes, they were drawn 400 points above the window border and 400 points to the right of the window border. The points are plotted in accordance with the dimensions of the window, not the axes. Since the axes has been shifted 400 spaces, the coordinates also have to be shifted.

RESULTS

The message explaining to the user what he can do by entering the program is displayed once as the program is first entered. The program also displays the message of not being able to read a file every time a "Response = -1." appears on the screen. The program successfully opened a storage file and set it up in my directory. I did, however, have to set the mode on the file again in order for me to be able to read and write to it. This was done by using the Utek command, "chmod". This file was sent to a diskette using the "cpio" command, thus making a hard copy of the test. The FORTRAN program "wave" did read the data points found in a disk file onto an array and plotted the points on an x-y coordinate system. By hitting a button found on the computer called "print screen", the graph was sent to the laser printer and a hard copy of the waveform was made. Hence, all my objectives for the summer have been fulfilled.

CONCLUSION

I have learned over the past two summers that not only computer programming, but all scientific research is accomplished through trial and error. I tried to make certain additions to the program a myriad of times only to find that these ideas were wrong. It took many tries to find out exactly what the computer would accept, but all I could do was keep trying until I found the right combination. To be successful in any field, I have discovered, one has to expect to fail before they can succeed.

BIBLIOGRAPHY

Plot 10 Standard Tektronix Interface Escapes Manual

Schildt, Herbet. C, The Complete Reference. Mcgraw-Hill: Berkley, Ca., 1987.

Utek Commands Manual. Vol.2.

PROGRAM LISTING

```
/*program gpibtst.f
* This simple program permits a user to send "set" and "query"
*commands to one of two instruments connected to /dev/gpiba.
* The primary benefit of this program is that it shows how to
* correctly poll one or more instruments when the auto-poll
* feature is enabled.
*
* It is assumed that the following GPIB instruments are attached
* to /dev/gpiba via gpconf(1) :
*
*       /dev/scope1 - a Tek Codes and Formats oscilloscope
*       /dev/scope2 - a Tek Codes and Formats oscilloscope
*
* It is also assumed that gpconf(1) has been used to enable
* auto-polling of these instruments.
*
* ------------------------------------------------------------------
*
* Usage:
*       This program prompts the user for newline-delimited input
*       from "stdin".  The first character of each input line is
*       examined and if it is "s" or "S", the input line (sans
*       first character) is sent to /dev/scope1 as a device-dependent
*       command.  Immediately thereafter, this program reads a
*       query response (if any) from /dev/scope1.
*
*       If the first character is "t" the same steps
*       as above are taken, but the input/output device is
*       /dev/scope2 instead.
*
*       If the letter "y" is typed in than the buffer r is written
*       to disk file "store".If the letter "x" is typed than the
*       program is existed.
"rtd.c" 127 lines, 4250 characters
```

```
*
* Caveats:
*       This program performs no I/O error checks.
*/

#include <stdio.h>
#include <signal.h>
#include <sys/file.h>
#include <sys/ioctl.h>
#include <box/gpibb.h>
#include <box/gpb_ioctl.h>
#define         BUFLEN  10000
int scope1,scope2;
main ()
{
        int bytes;
        int store;
        int t;
        char s[BUFLEN], r[BUFLEN];
        struct gpibconf cfg;
        void plot();
        void poll();
        scope1 = open("/dev/scope1", O_RDWR, 0);
        scope2 = open("/dev/scope2", O_RDWR, 0);
        ioctl(scope1, GIOCGCONF, &cfg);          /* get GPIB configure structure
*/
        cfg.gc_flags |= GF_ASYNC;                /* enable the signal flag */
        cfg.gc_mask |= GI_SRQ;
        cfg.gc_pgrp = getpgrp(0);                /* enable the signal process gro
up */
        ioctl(scope1, GIOCSCONF, &cfg);          /* set GPIB configure structure*
/
@
```

```
        ioctl(scope2, GIOCGCONF, &cfg);          /* get GPIB configure structure
*/
        cfg.gc_flags |= GF_ASYNC;                /* enable the signal flag */
        cfg.gc_mask |= GI_SRQ;
        cfg.gc_pgrp = getpgrp(0);                /* enable the signal process gro
up */
        ioctl(scope2, GIOCSCONF, &cfg);          /* set GPIB configure structure*
/
        signal(SIGURG, poll);                    /* Enable "fg" and "scope"
                                                  * interrupts.*/
        store = creat("/usr/schmidt/store", O_RDWR, 1);
        for(t=0;t!=1;++t){
              printf("You are now able to communicate with the digitizers.\n");
              printf("Please see RTD manual for a list of GPIB commands.\n");
              printf("Type s or S in front of each command to talk to\n");
              printf("scope1, and t to talk to scope2.\n");
              printf("If you want to make a plot of the waveform,type s,S,\n");
              printf("or t than PLOT.Than type y.\n");
              printf("If you wish to end this session type x.\n ");
        }

        for (;;) {
                printf("> ");

                gets(s);                         /* Read a GPIB command from stdi
n */
                if (s[0] == 's' || s[0] == 'S') {
                        write(scope1, &(s[1]), strlen(s) -1);
                        if(bytes = read(scope1, r, BUFLEN)==-1){
                        printf("Cannot read file.Please check to see if\n");
                        printf("scopes are on,or if all GPIB cables\n");
                        printf("are secured.\n");
                        }
```

```
                        r[bytes] = '*';
                        printf("Response = %d %s\n", bytes,r);
                        }
                        else if (s[0] == 't') {
                                write(scope2, &(s[1]), strlen(s) -1);
                                if(bytes = read(scope2, r, BUFLEN)==-1){
                                printf("Cannot read file.Please check to see if\n");
                                printf("scopes are on,or if all GPIB cables\n");
                                printf("are secured.\n");
                                }
                        r[bytes] = '*';
                        printf("Response = %d %s\n",bytes,r);                    /* Print
 response on stdout */
                        }
                        else if(s[0] =='y') {
                        write(store,r,BUFLEN);
                        }
                        else if(s[0] =='x'){
                        exit(0);
                        }
                        }
        }
void
poll()
{
        unsigned char status;
        /*
         * Poll each instrument on the bus. If bit 7 of an
         * instrument's status byte is set, that instrument
         * asserted SRQ.
         */
        ioctl(scope1, GIOCSPOLL, &status);              /* Serial poll */
        if (status & 0x40)                              /* Test bit 7 */
```

```
                        if(bytes = read(scope2, r, BUFLEN)==-1){
                        printf("Cannot read file.Please check to see if\n");
                        printf("scopes are on,or if all GPIB cables\n");
                        printf("are secured.\n");
                        }
                r[bytes] = '*';
                printf("Response = %d %s\n",bytes,r);                    /* Print
 response on stdout */
                }
                else if(s[0] =='y') {
                write(store,r,BUFLEN);
                }
                else if(s[0] =='x'){
                exit(0);
                }
                }
        }
void
poll()
{
        unsigned char status;
        /*
         * Poll each instrument on the bus. If bit 7 of an
         * instrument's status byte is set, that instrument
         * asserted SRQ.
         */
        ioctl(scope1, GIOCSPOLL, &status);              /* Serial poll */
        if (status & 0x40)                              /* Test bit 7 */
        printf("* scope1 status:   %d",status);
        ioctl(scope2, GIOCSPOLL, &status);              /* Serial poll */
        if (status & 0x40) /* Test bit 7 */
        printf("* scope2 status:   %d", status);
}
```

```
      program wave
c     ************************************************************
c     *                                                          *
c     *  This program creates an x-y graph using data from an input    *
c     *  file to be specified in subroutine array.  Each subroutine    *
c     *  documents its purpose and use.  In-line comments are also     *
c     *  provided.                                               *
c     *                                                          *
c     ************************************************************
      common xarray(164), yarray(164), code1, xint, yint
      character*1 code, code1
      integer iview
      iview = 0
c
c     Determine when to start and stop plotting
c
      call llinit(jterm)
      call lllnwd(2)
      call llhcin(1)
      call llhcid(1)
      call llhcda(2)
      call llhcdt(3)
      call llhcsz(0)
      call llcode(0)
      call llpage
c
c     Begin actual plotting
c
      call tinit
      call array
      call axes
      call graph
c
```

```
c
c       Now waiting for signal to clear screen
c
        call llstop
        call llecho(1)
        end
c
c
        subroutine tinit
c
c       This subroutine initializes the terminal parameters needed.
c
        integer isurf
        isurf=4
        call lldavs(0)
        call lldacl
        call llvwat(1,0,1)
        call llclmd(3,1,1)
        call llerth(3)
        call lltxin(32333)
        call llvwpt(0,0,4095,3276)
        call llwind(0,0,4095,3276)
        call lldfsf(1,4)
        call llrnvw(-1)
        call lldavs(1)
        return
        end
c
c
        subroutine array
c
c       This subroutine opens the data file and loads the data into an
c       array for x and an array for y values.  For different length data
```

```
c       files, make sure that all common dimension statements are adjusted.
c
        common xarray(164), yarray(164), code1, xint, yint
        open(5,file='store1',status='old')
        xmax=0.0
        ymax=0.0
        do 10 i=1,164
        call lldump
        read(5,*) x,y
        xarray(i)=x
        yarray(i)=y
        if(x.gt.xmax)xmax=x
        if(y.gt.ymax)ymax=y
   10   continue
        xfactor=3600./xmax
        yfactor=2800./ymax
        do 20 j=1,164
        xarray(j)=xarray(j)*xfactor
        yarray(j)=yarray(j)*yfactor
   20   continue
        xint=xmax/10.0
        yint=ymax/10.0
        return
        end
c
c
        subroutine axes
c
c       This subroutine draws the vertical and horizontal axes of the graph
c       and puts tic marks on the axes.
c
c       Here the axes are being drawn
c
```

```
c
      common xarray(164), yarray(164), code1, xint, yint
      character*8 ixlab
      character*4 iylab
      integer xlab,ylab
      call llmove(4000,406)
      call llinin(1)
      call lldraw(200,400)
      call llmove(400,3200)
      call llinin(1)
      call lldraw(400,200)
      call llmove(0,0)
c
c     Putting the tic marks on the axes.  Alter the do stmt to change the
c     interval
c
      j=1
      do 20 l=760,4000,360
      call llmove(l-360,200)
      xlab=xint*j
      write(ixlab,100) xlab
  100 format(i8)
      call llgsiz(30,45,30)
      call lltext(8,ixlab)
      call llmove(l,400)
      call llinin(4)
      call lldraw(l,300)
      j=j+1
   20 continue
c
      k=1
      do 30 j=680,3200,280
      call llmove(0,j-20)
```

```
c
      common xarray(164), yarray(164), code1, xint, yint
      character*8 ixlab
      character*4 iylab
      integer xlab,ylab
      call llmove(4000,400)
      call lllnin(1)
      call lldraw(200,400)
      call llmove(400,3200)
      call lllnin(1)
      call lldraw(400,200)
      call llmove(0,0)
c
c     Putting the tic marks on the axes.  Alter the do stmt to change the
c     interval
c
      j=1
      do 20 l=760,4000,360
      call llmove(l-360,200)
      xlab=xint*j
      write(ixlab,100) xlab
  100 format(i8)
      call llgsiz(30,45,30)
      call lltext(8,ixlab)
      call llmove(l,400)
      call lllnin(4)
      call lldraw(l,300)
      j=j+1
   20 continue
c
      k=1
      do 30 j=680,3200,280
      call llmove(0,j-20)
```

```
      call llgsiz(25,40,25)
      ylab=yint*k
      write(iylab,200) ylab
  200 format(i4)
      call lltext(4,iylab)
      call llmove(400,j)
      call lllnin(4)
      call lldraw(300,j)
      k=k+1
   30 continue
      return
      end
c
c
      subroutine graph
c
c     This subroutine reads the data from the arrays and plots the points in
c     'connect-the-dot' fashion.  The positions are altered by adding 200
c     to data value to adjust the absolute postion of the data point to
c     match the axes.
c
      common xarray(164),yarray(164)
      integer x,y
      x=xarray(1)+400
      y=yarray(1)+400
      call llmove(x,y)
      do 10 i=2,164
      x=xarray(i)+400
      y=yarray(i)+400
      call lllnin(5)
      call lldraw(x,y)
   10 continue
      return
```

```
      write(iylab,200) ylab
  200 format(i4)
      call lltext(4,iylab)
      call llmove(400,j)
      call lllnin(4)
      call lldraw(300,j)
      k=k+1
   30 continue
      return
      end
c
c
      subroutine graph
c
c     This subroutine reads the data from the arrays and plots the points in
c     'connect-the-dot' fashion.  The positions are altered by adding 200
c     to data value to adjust the absolute postion of the data point to
c     match the axes.
c
      common xarray(164),yarray(164)
      integer x,y
      x=xarray(1)+400
      y=yarray(1)+400
      call llmove(x,y)
      do 10 i=2,164
      x=xarray(i)+400
      y=yarray(i)+400
      call lllnin(5)
      call lldraw(x,y)
   10 continue
      return
      end
```

CURRENT SIMULATIONS IN ELECTROMAGNETIC LAUNCHER POWER SUPPLIES

BY

PATRICIA TU

MARK HEYSE, MENTOR

EGLIN AIR FORCE BASE
HIGH SCHOOL APPRENTICESHIP PROGRAM 1990

ACKNOWLEDGEMENTS

I would like to thank

Mark Heyse, my mentor, for teaching me about electromagnetic launchers and their power supplies and answering all my questions concerning them.

Jeff Gay for all that he has taught me about circuits and for allowing me to use his Macintosh to write my report.

Toni Audi, Don Littrell, and the many other people whom I have received computer programming advice from.

Mr. Don Harrison, Dr. Norm Klausutis, Ms. Ellen Long and the other people involved with making this program possible.

The people at Site A-15 for their friendship and support.

The other apprentices at Eglin for their friendship.

The Air Force Armament Laboratory's Hypervelocity Launcher Research Division (SAH) has been established to research electromagnetic launchers (EMLs) or rail guns. This research is funded by SDI and is seen as a potential method of defending the United States against ballistic missiles and reentry vehicles. These rail guns have the potential to launch guided projectiles at velocities up to 10 km/sec. The power sources of these guns need to be able to store high levels of energy and discharge it to the gun in a small amount of time. Many power sources such as the battery power supplies and the 5 MegaJoule capacitor system have been built on site to provide the necessary high current levels.

During the summer I was involved with battery power supplies. One system, the Prototype Battery Power Supply (PBPS), is composed of 960 car batteries. This system is being expanded to contain a total of 1440 batteries and will be able to produce up to 250,000 Amps. A program is needed to determine if the higher currents can be reached after the system is expanded.

The PBPS system is a modulator design that centers on the use of standard automotive batteries. Figure 1 shows one tray which is 10 batteries in series. Each tray of batteries is 127 volts. Two trays make a string and there are 72 strings in this system. The 72 strings of batteries can be set up in four different configurations to reach the desired current levels. One configuration is to have up to 72 strings in parallel (Figure 2 A). Another is to have up to 72 strings in parallel with the addition of grading resistors (Figure 2 B). The two other possible configurations are to use up to 72 strings in series

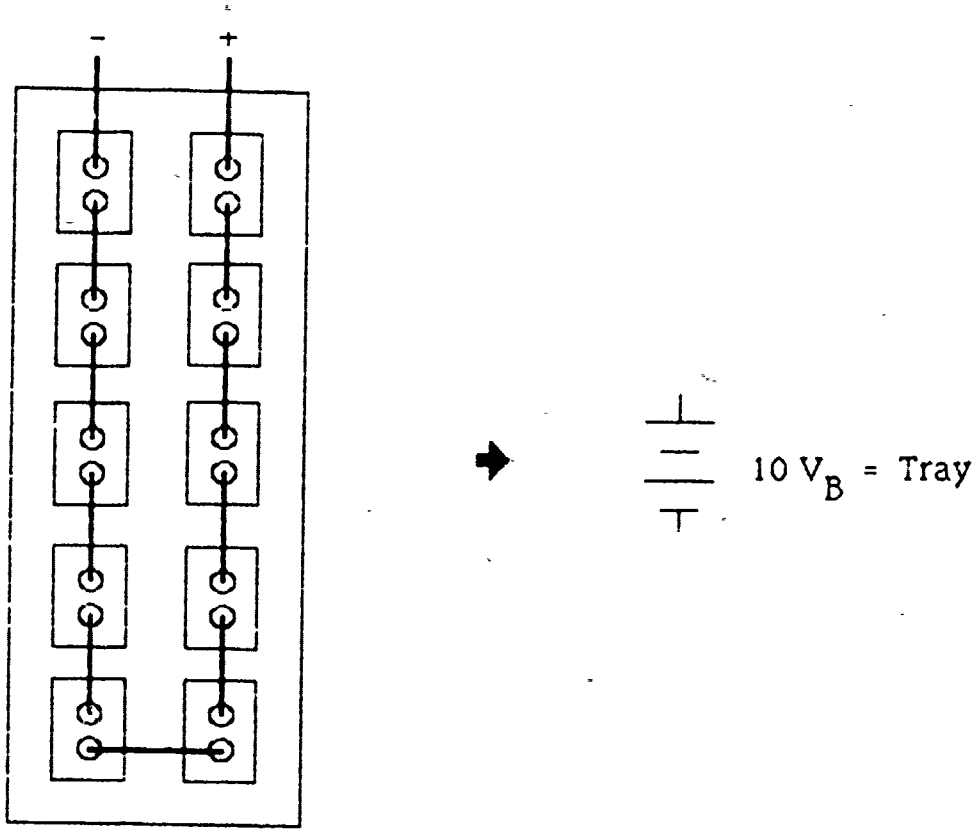


Figure 1 10 batteries in series = 1 tray

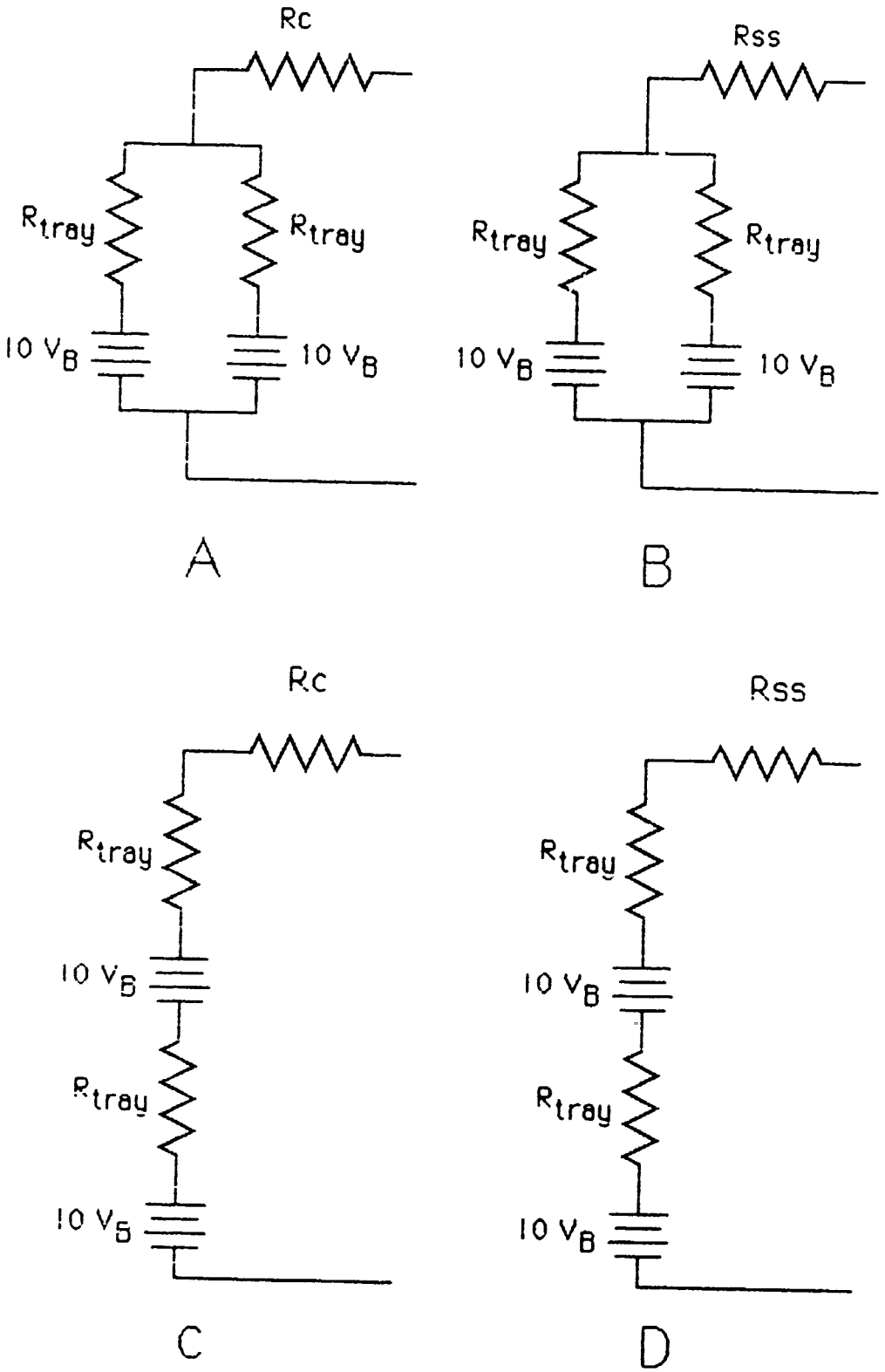


Figure 2 String Configurations

either with or without grading resistors (Figure 2 C & D). Rc is the resistance of the copper cable when the stainless steel grading resistors (Rss), are bypassed. The copper cable is used to connect the strings directly to the gang switches. Grading resistors are solid stainless steel rods which lowers the current output of the batteries.

Grading Resistors are needed because a battery has a 2000 A limit so only 4000 Amps can be pulled safely from a string of batteries in parallel. When the batteries are in series only 2000 Amps can be pulled from a string. In the cases where the desired current level can be reached without exceeding the limitations of the batteries the grading resistors are not necessary. In that case a switch will allow the current to flow through the copper cable instead of the grading resistors.

Figure 3 shows how the 72 strings of batteries in the system are set up. N is the number of strings of batteries from 1 to 72. The individual string currents, Is, are summed to determine the total system current, It. Rload is the resistance in the gang switch, buss work, storage inductor, opening switch, and any other external resistance.

A program called "Battery Configurations" was written to determine all possible system currents and string currents for all four string configurations. Figure 4 is a flowchart for this program. Since the internal resistance is slightly different for each configuration this program solves first for the internal resistance then adds that to the constant load resistance. After the total resistance is found the total current and current per string is

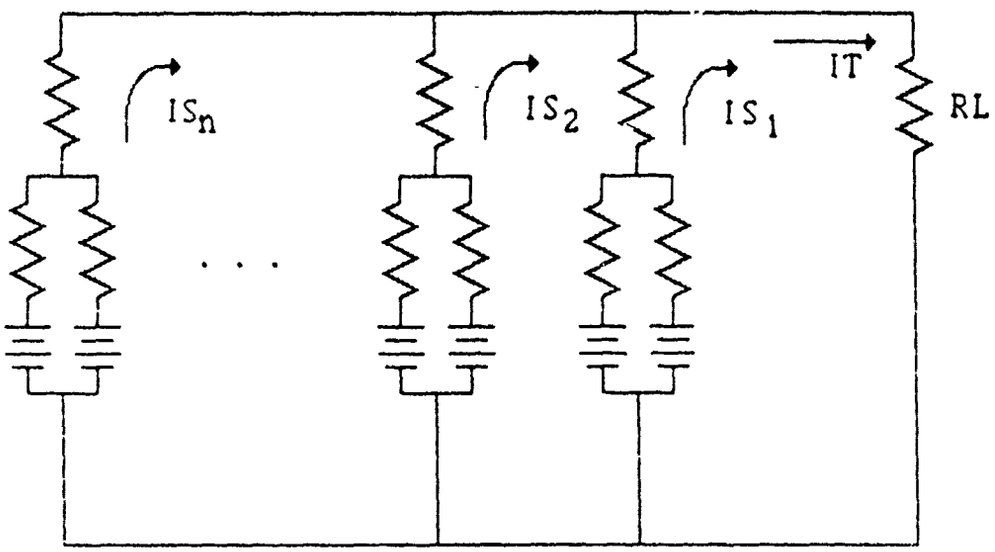


Figure 3 System Configuration

BATTERY CONFIGURATIONS

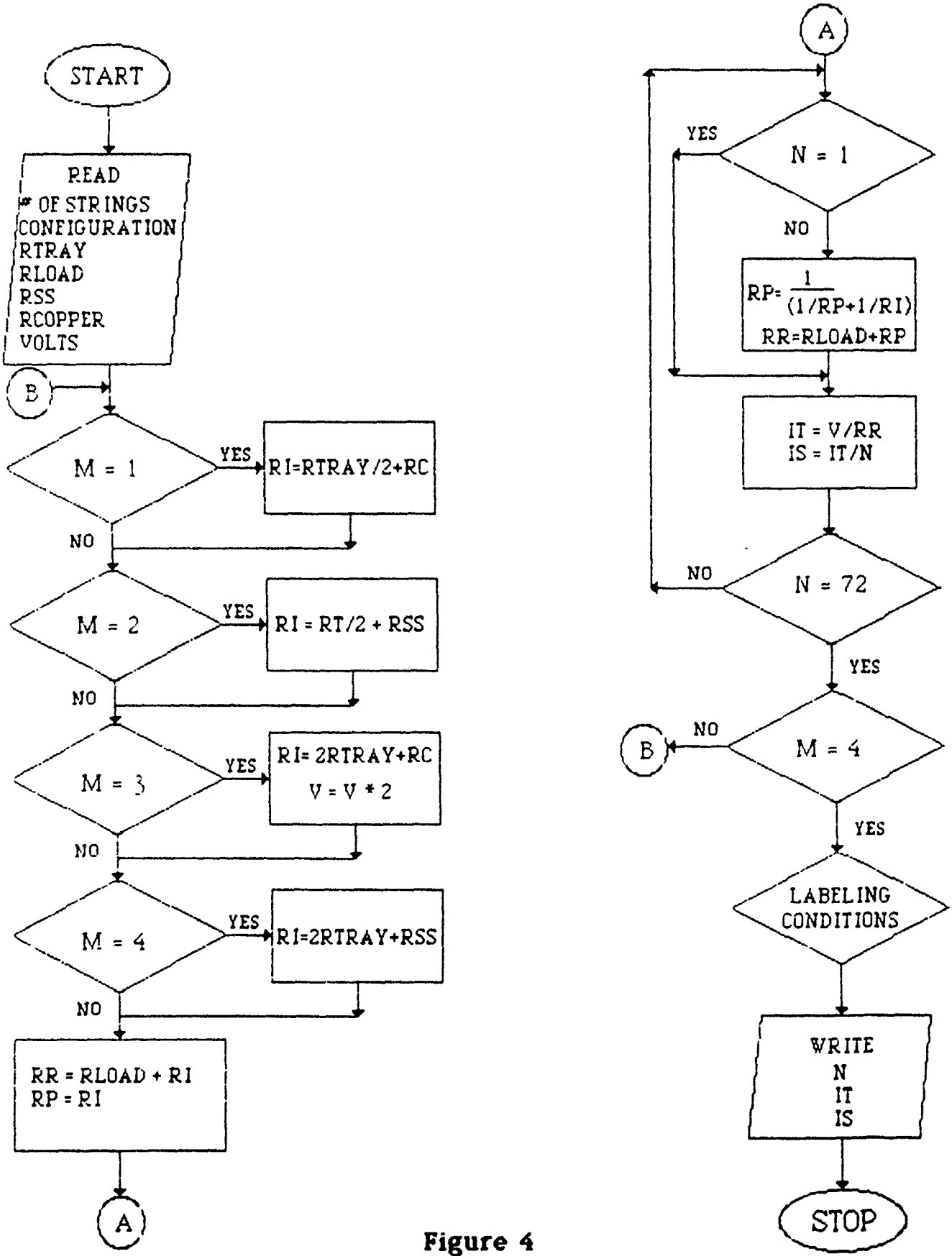


Figure 4

calculated. The simulation starts with one string of batteries and continues adding strings to include all 72 strings. The new resistance of each string is added in parallel to the previous resistance as the strings of batteries are added. After all 72 strings of one configuration are calculated the calculations for the next configuration begins. The output is shown in as Table 1 and Table 2. Table 1 shows the total current of all the strings being used in Amps. Table 2 shows the current per string in Amps. To use these charts the current level that is desired is found on the total current chart and the current per string chart is checked to ensure that the current out of each string of batteries is not exceeding its established limits. These are plotted for easier understanding in Figure 5 and Figure 6.

It can be seen when studying the graphs of the series without grading resistors configuration that the current produced ranged from 2700 Amps to 186,000 Amps. However, each string is producing more than 2000 Amps. Since the strings were pulling more than its limit this configuration can not be used safely.

In the parallel with grading resistor configuration and series with grading resistor configuration the current levels produced are approximately 1,980 Amps to 140,000 Amps. The current per string is also within its limits. Therefore these configurations can be used to reach the stated current levels.

In the parallel without grading resistors configuration the current levels are high and also desirable. This is the only configuration that produces currents over 200,000 Amps and these

TABLE 1

TOTAL CURRENT IN AMPS

	PARALLEL W/O GR	PARALLEL WITH GR	SERIES W/O GR	SERIES WITH GR
1	4817.357	2096.990	2706.072	1983.399
2	9611.745	4189.622	5408.514	3964.847
3	14383.33	6277.908	8107.333	5944.348
4	19132.27	8361.864	10802.54	7921.904
5	23858.72	10441.50	13494.13	9897.517
6	28562.86	12516.83	16182.13	11871.19
7	33244.83	14587.88	18866.53	13842.93
8	37904.79	16654.64	21547.34	15812.74
9	42542.89	18717.15	24224.57	17780.61
10	47159.30	20775.40	26898.23	19746.56
11	51754.16	22829.41	29568.33	21710.58
12	56327.62	24879.20	32234.87	23672.68
13	60879.83	26924.77	34897.86	25632.86
14	65410.94	28966.15	37557.30	27591.13
15	69921.09	31003.34	40213.21	29547.48
16	74410.44	33036.36	42865.59	31501.92
17	78879.12	35065.21	45514.45	33454.45
18	83327.27	37089.92	48159.78	35405.08
19	87755.04	39110.50	50801.61	37353.80
20	92162.57	41126.95	53439.94	39300.63
21	96550.00	43139.29	56074.78	41245.56
22	100917.5	45147.54	58706.13	43188.60
23	105265.1	47151.70	61333.99	45129.74
24	109593.0	49151.79	63958.39	47069.00
25	113901.4	51147.81	66579.31	49006.37
26	118190.3	53139.79	69196.77	50941.85
27	122459.9	55127.74	71810.78	52875.46
28	126710.4	57111.66	74421.34	54807.19
29	130941.8	59091.57	77028.48	56737.04
30	135154.3	61067.49	79632.16	58665.03
31	139348.1	63039.42	82232.43	60591.14
32	143523.1	65007.37	84829.27	62515.38
33	147679.7	66971.36	87422.70	64437.76
34	151817.8	68931.40	90012.73	66358.27
35	155937.6	70887.50	92599.36	68276.94
36	160039.2	72839.68	95182.59	70193.75
37	164122.8	74787.94	97762.44	72108.70
38	168188.5	76732.30	100338.9	74021.80
39	172236.4	78672.76	102912.0	75933.06
40	176266.5	80609.35	105481.7	77842.47
41	180279.1	82542.06	108048.1	79750.03
42	184274.2	84470.92	110611.1	81655.77
43	188252.0	86395.94	113170.8	83559.66
44	192212.5	88317.13	115727.1	85461.71
45	196155.9	90234.48	118280.1	87361.94
46	200082.2	92148.03	120829.8	89260.34
47	203991.7	94057.78	123376.1	91156.91
48	207884.4	95963.75	125919.2	93051.65
49	211760.4	97865.94	128458.9	94944.57
50	215619.8	99764.36	130995.4	96835.68
51	219462.7	101659.0	133528.5	98724.96
52	223289.3	103550.0	136058.3	100612.4
53	227099.5	105437.1	138584.9	102498.1
54	230893.7	107320.6	141108.2	104382.0
55	234671.7	109200.4	143628.3	106264.0
56	238433.8	111076.4	146145.0	108144.3
57	242180.0	112948.8	148658.5	110022.7
58	245910.5	114817.5	151168.8	111899.4
59	249625.3	116682.5	153675.8	113774.2
60	253324.5	118543.9	156179.6	115647.3
61	257008.3	120401.6	158680.2	117518.6
62	260676.7	122255.7	161177.5	119388.1
63	264329.8	124106.2	163671.6	121255.7
64	267967.8	125953.0	166162.5	123121.7
65	271590.7	127796.3	168650.2	124985.8
66	275198.6	129636.0	171134.7	126848.2
67	278791.6	131472.0	173616.0	128708.8
68	282369.8	133304.5	176094.1	130567.6
69	285933.3	135133.5	178569.0	132424.6
70	289482.2	136958.9	181040.8	134279.9
71	293016.5	138780.8	183509.3	136133.4
72	296536.4	140599.1	185974.8	137985.2

TABLE 2

CURRENT PER STRING IN AMPS

	PARALLEL W/O GR	PARALLEL WITH GR	SERIES W/O GR	SERIES WITH GR
1	4817.357	2096.990	2706.072	1983.399
2	4805.873	2094.811	2704.257	1982.423
3	4794.443	2092.636	2702.444	1981.449
4	4783.067	2090.466	2700.634	1980.476
5	4771.745	2088.300	2698.826	1979.503
6	4760.476	2086.139	2697.021	1978.532
7	4749.261	2083.983	2695.218	1977.562
8	4738.099	2081.831	2693.417	1976.592
9	4726.988	2079.683	2691.619	1975.624
10	4715.930	2077.540	2689.823	1974.656
11	4704.924	2075.401	2688.030	1973.689
12	4693.968	2073.266	2686.239	1972.724
13	4683.064	2071.136	2684.451	1971.759
14	4672.210	2069.010	2682.665	1970.795
15	4661.406	2066.889	2680.881	1969.832
16	4650.652	2064.772	2679.099	1968.870
17	4639.948	2062.660	2677.320	1967.909
18	4629.292	2060.551	2675.543	1966.949
19	4618.686	2058.447	2673.769	1965.990
20	4608.128	2056.347	2671.997	1965.032
21	4597.619	2054.252	2670.228	1964.074
22	4587.157	2052.161	2668.460	1963.118
23	4576.743	2050.074	2666.695	1962.163
24	4566.375	2047.991	2664.933	1961.208
25	4556.055	2045.912	2663.173	1960.255
26	4545.781	2043.838	2661.414	1959.302
27	4535.554	2041.768	2659.658	1958.350
28	4525.372	2039.702	2657.905	1957.400
29	4515.235	2037.641	2656.154	1956.450
30	4505.145	2035.583	2654.406	1955.501
31	4495.099	2033.530	2652.659	1954.553
32	4485.098	2031.480	2650.915	1953.606
33	4475.141	2029.435	2649.173	1952.659
34	4465.229	2027.394	2647.433	1951.714
35	4455.360	2025.357	2645.696	1950.770
36	4445.534	2023.324	2643.961	1949.826
37	4435.752	2021.296	2642.228	1948.884
38	4426.013	2019.271	2640.498	1947.942
39	4416.317	2017.250	2638.770	1947.002
40	4406.663	2015.234	2637.043	1946.062
41	4397.051	2013.221	2635.320	1945.123
42	4387.481	2011.212	2633 598	1944.185
43	4377.953	2009.208	2631.879	1943.248
44	4368.465	2007.207	2630.162	1942.312
45	4359.019	2005.211	2628.447	1941.376
46	4349.614	2003.218	2626.735	1940.442
47	4340.250	2001.229	2625.024	1939.509
48	4330.925	1999.245	2623.316	1938.576
49	4321.640	1997.264	2621.611	1937.644
50	4312.396	1995.287	2619.907	1936.714
51	4303.190	1993.314	2618.206	1935 784
52	4294.024	1991.345	2616.507	1934.855
53	4284.896	1989.380	2614.810	1933.927
54	4275.809	1987.419	2613.115	1932.999
55	4266.758	1985.461	2611.423	1932.073
56	4257.746	1983.508	2609.733	1931.148
57	4248.772	1981.558	2608.044	1930.223
58	4239.835	1979.612	2606.359	1929.299
59	4230.937	1977.670	2604.675	1928.377
60	4222.075	1975.732	2602.993	1927.455
61	4213.251	1973.797	2601.314	1926.534
62	4204.463	1971.866	2599.637	1925.614
63	4195.712	1969.939	2597.962	1924.694
64	4186.997	1968.016	2596.289	1923.776
65	4178.318	1966.097	2594.618	1922.859
66	4169.676	1964.181	2592.949	1921.942
67	4161.068	1962.269	2591.283	1921.026
68	4152.497	1960.361	2589.619	1920.111
69	4143.961	1958.456	2587.957	1919.197
70	4135.460	1956.556	2586.296	1918.284
71	4126.993	1954.658	2584.639	1917.372
72	4118.562	1952.765	2582.983	1916.460

Figure 5 Total Current

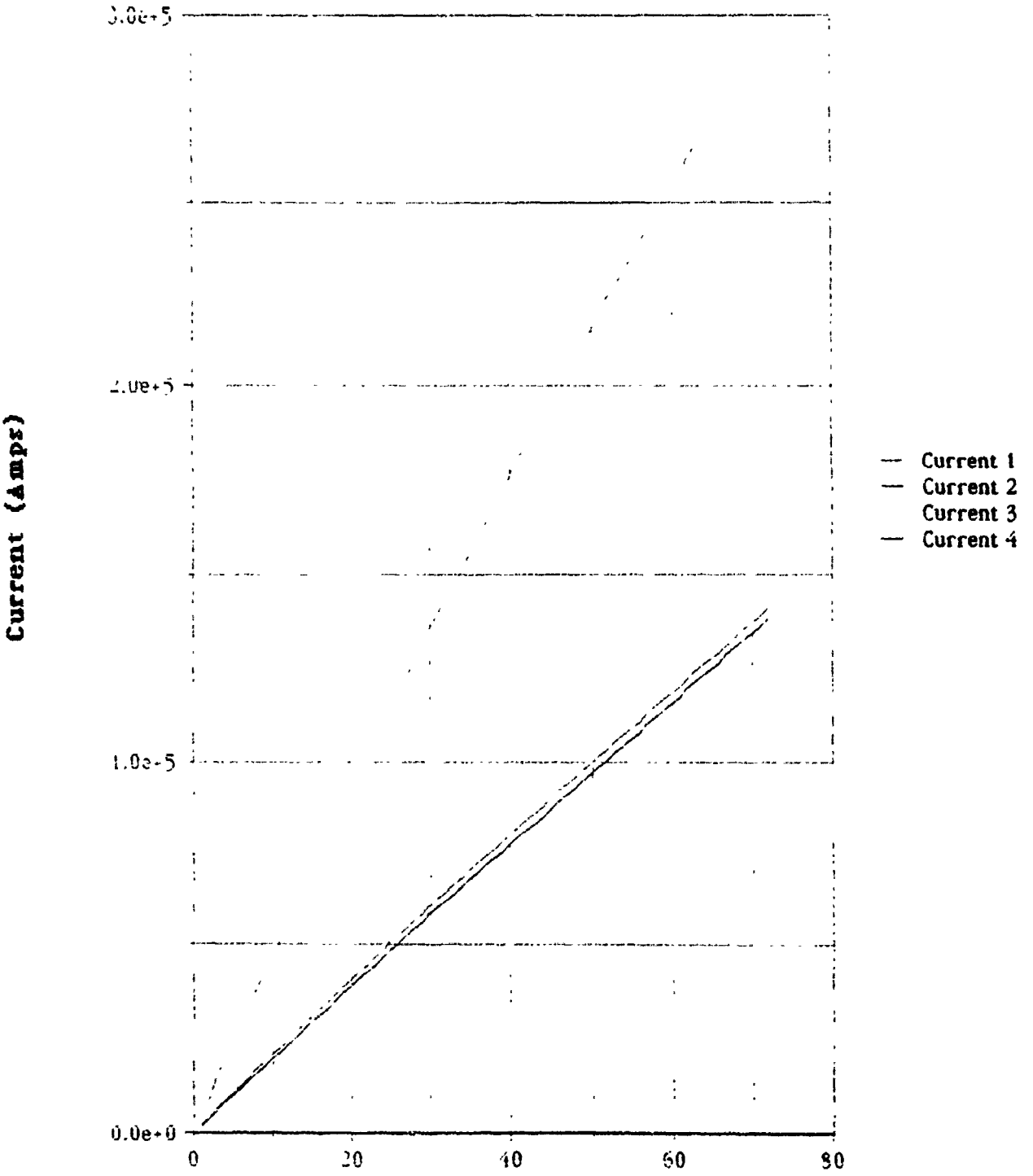
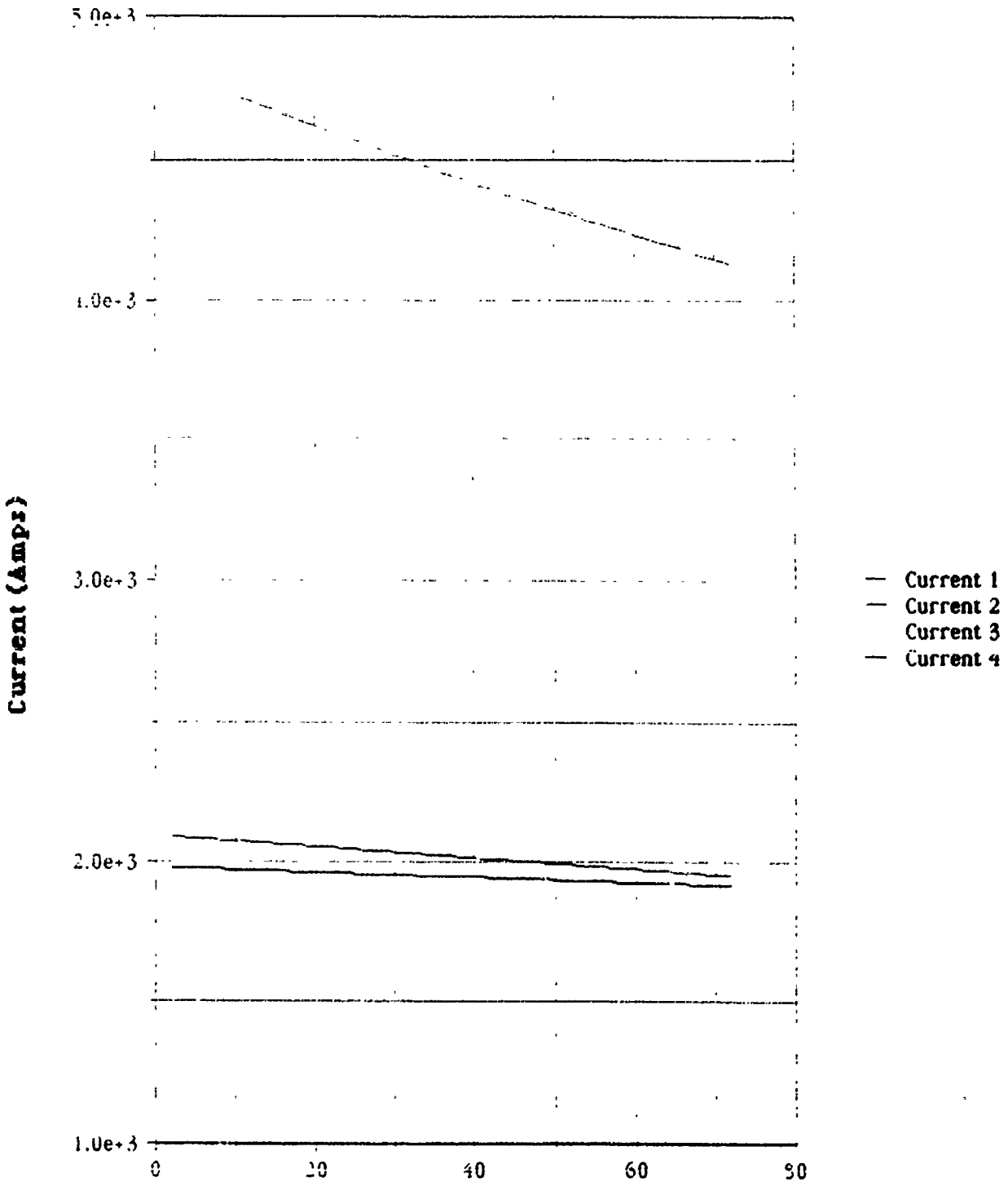


Figure 6 Current Per String



currents are wanted. The problem is that each string is producing over 4000 Amps.

To solve this problem external resistance can be added to the system. Adding resistance will still produce the high current levels, but it will also lower the current per string to a safer level. Since this is a computer program the external resistance can be added and changed easily in the attempts to achieve the best results. To safely reach 200,000 Amps a resistance of 35 microOhms needs to be added to the original internal resistance. Figure 7 and Figure 8 show the results of this change.

FIGURE 7 200,000 AMPS

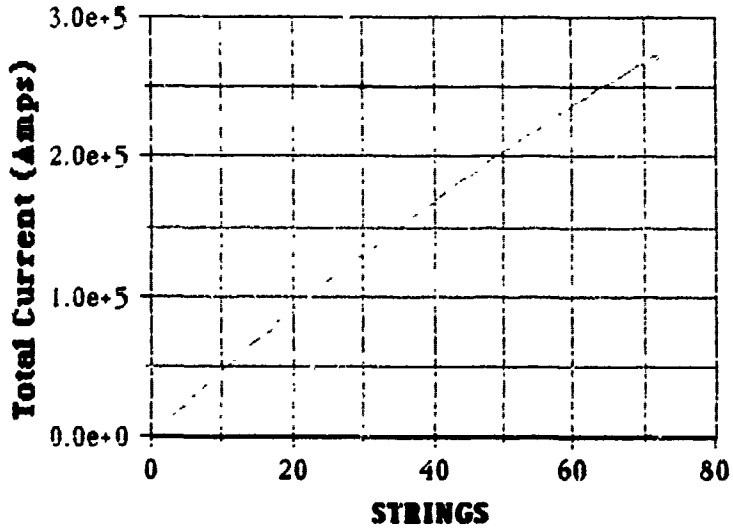
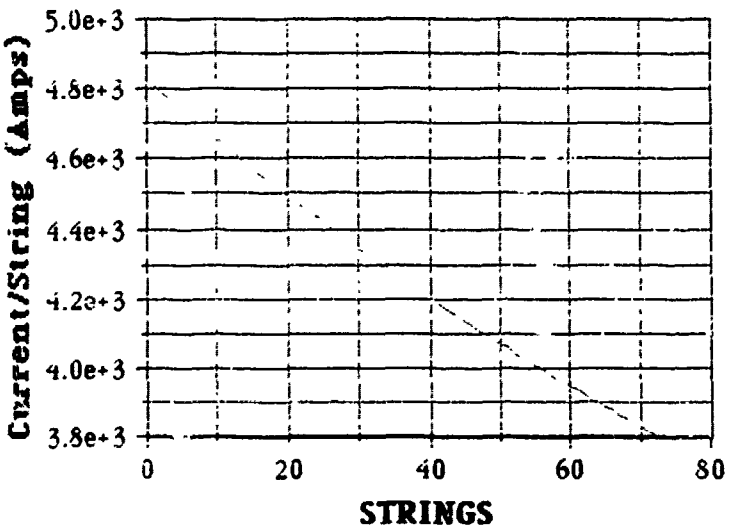


Figure 8 200,000 Amps



To safely reacn 250,000 Amps a resistance of 10 microOhms needs to be added to the original resistance. Figure 9 and Figure 10 show the successful results of this change.

Figure 9 250,000 Amps

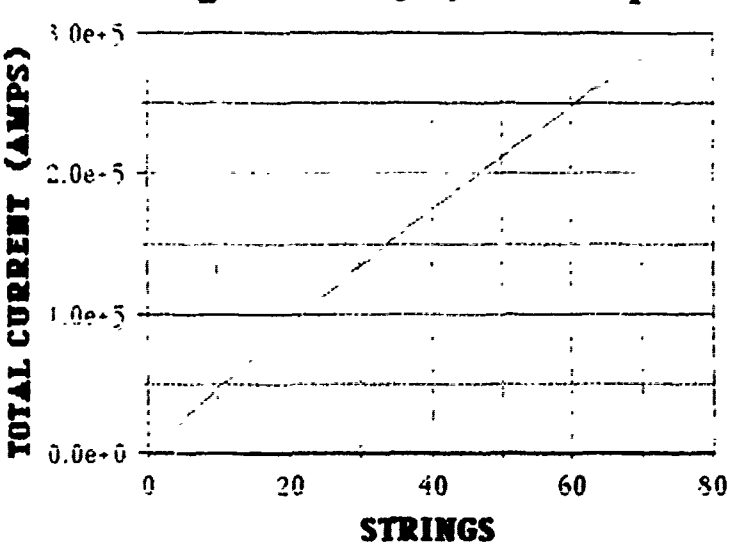
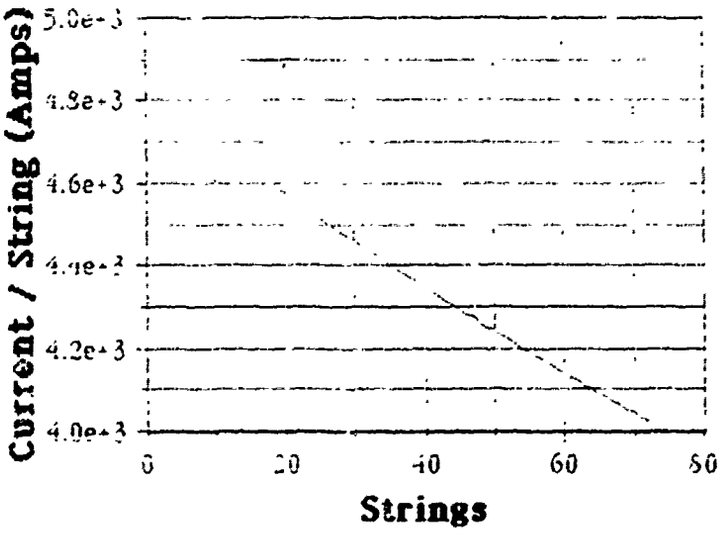


Figure 10 250,000 Amps



Now it is known that 50 KA to 140 KA can be reached easily. It is also known that by adding resistances to the system the higher current levels can be reached safely. The users of the system will decide if the changes are possible and practical.

This program has been expanded to identify the string configuration for a specified current level. All the possible configurations are displayed. The user chooses the configuration that stresses the batteries the least. This battery configuration program and an example of the output when the desired current is 150 KA is included in Appendix A. Appendix B is the program used when plotting.

With this program it can be seen that from 0 - 140,000 Amps of current can be easily reached, but there is a window from 150,000 Amps - 250,000 Amps that can not be reached safely. To reach those higher current levels external resistance can be added to the

system to lower the current output per string to a safe level. Through the use of this program the amount of resistance can be determined.

Another system is the Multishot Battery Power Supply which is made of 240 car batteries. It produces up to 12,000 Amps and is used to charge a 100,000 Joule capacitor bank. In this system Ross relays are used. These relays are difficult to control and at times may weld shut. They are to be replaced with solid state devices. However, if solid state devices are to be used the switching requirements and device limitations must be known. A program which simulates the current with respect to time is written to determine the switching requirements.

The Multishot Battery Power Supply is a system that is the power source of the rapid fire guns. These batteries are used to charge the capacitors which are then discharged to the gun. As with the other power supplies all the batteries need not be used at once, but a tray is the smallest unit to be used. Six trays make a bank and there are 4 banks. Each bank produces up to 760 Volts. The intended resistance and capacitance for each bank has been calculated as 0.240 Ohms and 0.084 Farads, respectively. The inductance for 60 batteries is estimated to be 10 microHenries. Figure 11 is this simple series RLC circuit.

To find the current as time changed in this RLC circuit the following equation was used.

$$I(t) = A_1 e^{s1t} + A_2 e^{s2t}$$

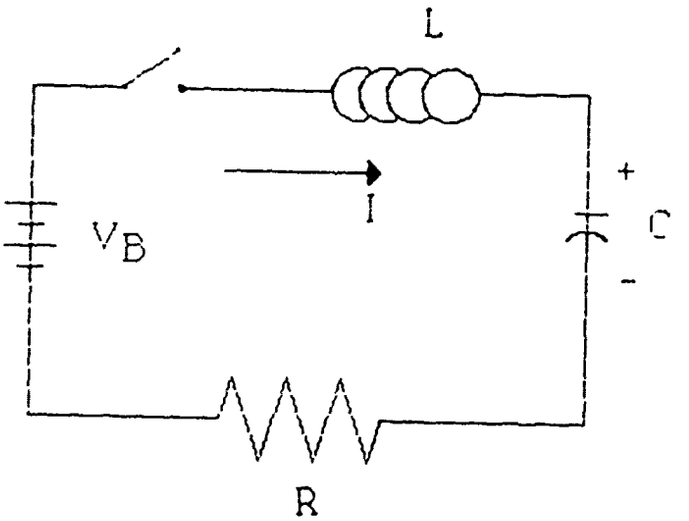


Figure 11 Simple Series RLC Circuit

where

$$\alpha = R / 2C$$

$$\omega = 1 / SQR(LC)$$

$$S1 = -\alpha + SQR(\alpha^2 - \omega^2)$$

$$S1 = -\alpha - SQR(\alpha^2 - \omega^2)$$

and

$$A = V / L(S1 - S2)$$

A current level was calculated for the first 80 mS. It was noticed that the current peaked before 1mS so the peak current was not seen on the plots. Another loop was added to the program to calculate the current starting at .1mS to 1mS with a time step of .1mS then calculations were continued at every millisecond. The program flowchart is shown in Figure 12.

When the current is plotted with respect to time many interesting things happen as external resistance is changed. As more resistance is added the current peak is lowered but the total current falls slower. It was also noticed that as resistance was increased the current peaked more rapidly. An example of this is in Figure 13 . The external resistance, Rx, is changed from 0 Ohms, to .060 Ohms, and to .120 Ohms. The current peaks and time of current peaks are as follows:

CURRENT VS. TIME

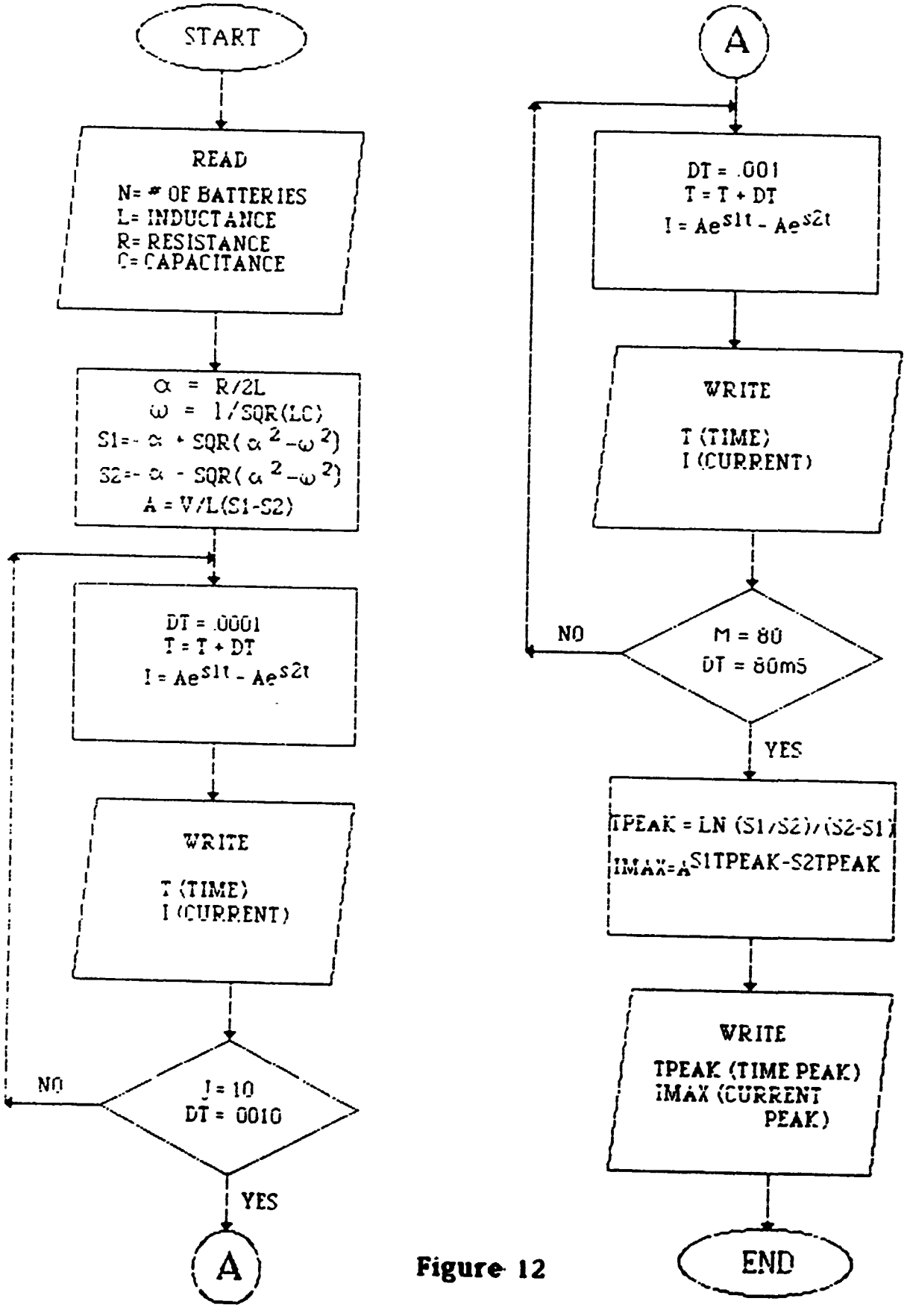
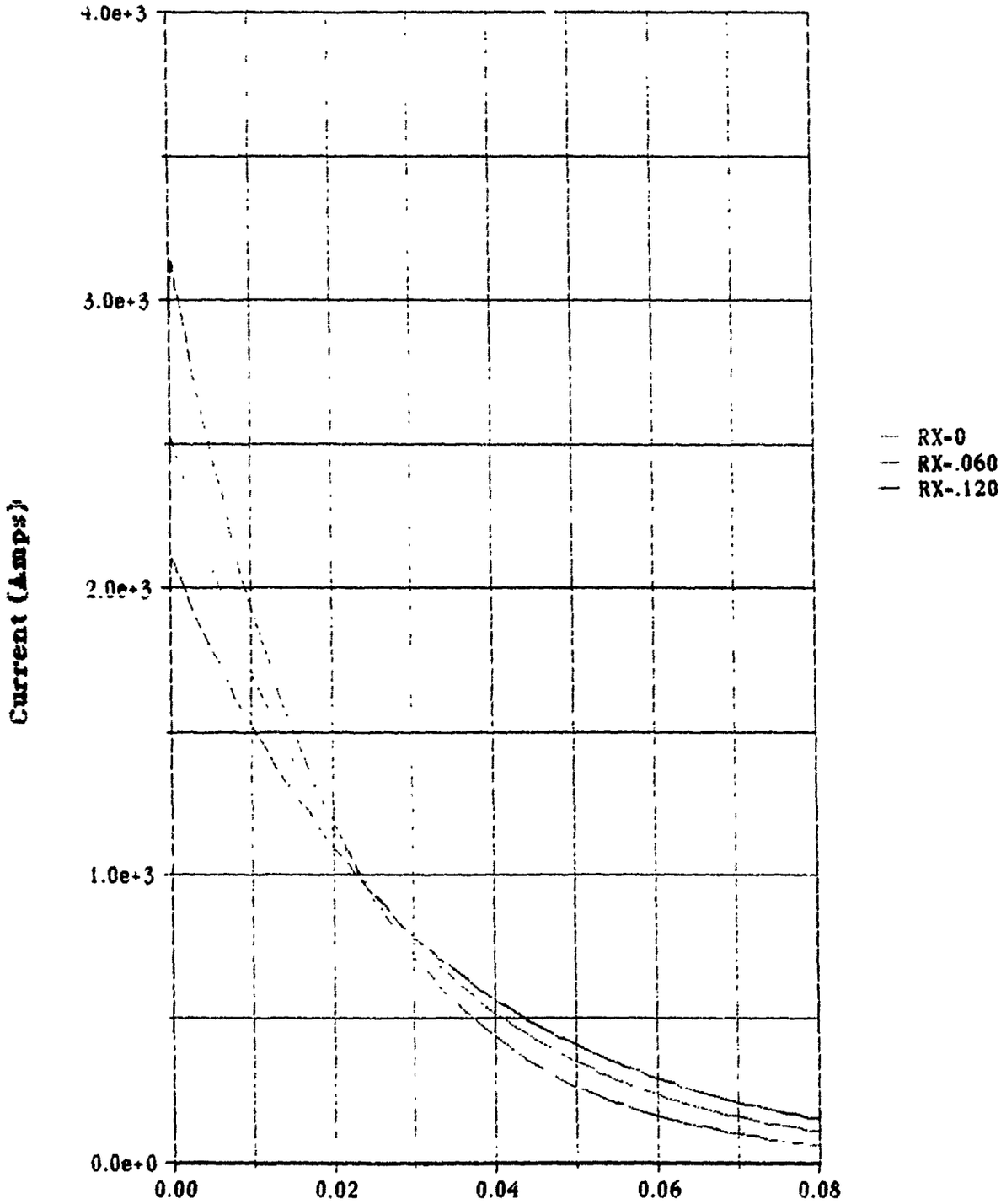


Figure 12

Figure 13 Charge Current



	CURRENT PEAK	TIME TO PEAK
RX = 0 Ohms	3133.55 Amps	.2585 mS
RX = .060 Ohms	2515.13 Amps	.2214 mS
RX = .120 Ohms	2100. Amps	.1946 mS

This program answers many questions about this power system. It is now known how much external resistance needs to be added to lower the current peak. It is also known that by adding inductance the time to peak is longer, but the peak current will only lower by approximately 100 Amps.

It was determined that the solid state device needs to be able to switch at 3000 Amps. If resistance is added it only needs to switch at 2000 Amps. Many different cases were run with different variables to determine the changes that needed to be made. This program and an example of the output is included in Appendix C.

The current simulation programs proved what was expected with the addition of batteries in the prototype battery power supply and showed what was required of the solid state devices in the multishot battery power supply.

I have learned a great deal at the Hypervelocity Launcher Technology Branch this summer. I have written programs about the current output of the prototype battery power supply and the multishot battery power supply. To write theses programs I had to learn about each power supply, how they are set up, how they operate, car batteries, circuits, and how to program in Fortran on the

VAX/VMS system. I also had the opportunity to learn about electrical circuits, their components, and how to make circuit boards. I increased my knowledge about electromagentic launchers and computers. I also learned that a lot of research has to be done before a project can begin.

BIBLIOGRAPHY

Didday, Rich; Page, Rex. **Fortran for Humans, Third Edition.** West Publishing Company. St. Paul, MN. 1981.

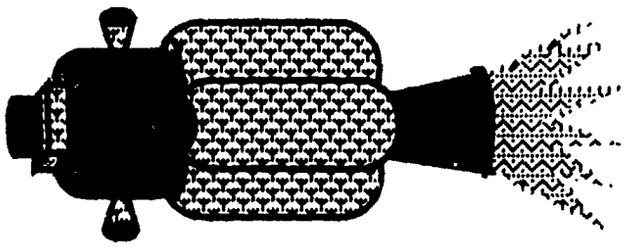
Lt. Gay, Jeff. **Personal Assistance**. Eglin Air Force Base. 1990.

Hayt, William, Jr.; Kemmerly, Jack E. **Engineering Circuit Analysis, Third Edition.** McGraw-Hill Book Company. NY,NY. 1990.

Mr. Heyse, Mark. Mentor. **Lectures**. Eglin Air Force Base. 1990.

Appendices can be obtained from
UNIVERSAL ENERGY SYSTEMS, INC.

Advanced Signal Processing Operations for Guided Interceptors



AFATL/SAI

Troy Urquhart
Fort Walton Beach High School
Mentor: Ms. Starla Christakos

Report Outline:

I. Background

II. The Hough Transform

III. The Multiple Image Signal Processor

A. Goals

B. Breakdown by Unit

IV. Acknowledgements

I. Background

This summer, I worked under Universal Energy Systems of Dayton, Ohio, for the Air Force Armament Laboratory on Eglin Air Force Base. More specifically, I worked in the Signal Processing section of SAI, the Guided Interceptor Technology branch of the Analysis and Strategic Defense Division.

SAI falls under the Strategic Defense Initiative Organization (SDIO), that organization which is ruthlessly hounded by the press and commonly called "Star Wars." The basic goal of SDI, as set forth in SAI's Introductory Package is four-fold:

1. Support a better basis for deterring aggression;
2. Strengthen strategic stability;
3. Increase the security of the United States and its Allies;
4. Eliminate the threat posed by ballistic missiles.

The Guided Interceptor branch deals with the design of (can you guess?) guided interceptors to meet these ends.

II. The Hough Transform

For the Hough transform, I took Derek Holland's basic ideas and molded them into something that actually worked!! The whole point of the Hough is to take a plume and extract the leading curve from which you can calculate an aimpoint to the hardbody. Well, Derek and I never got THAT far, but we did extract the leading curve pretty well.

The Hough starts out by playing a massive game of "Connect - the - dots" -- it connects every pixel in the image with every other pixel. It redefines these lines from a set of (x, y) to a set of (m, b) and plots them on the slope-intercept plane. Then, it counts the number of intersections at each point, thresholds to a given percentage of the maximum, and cross-references the remaining slopes and intercepts with the original image.

Basically, the Hough worked; the only problem I can see with it is that it takes almost forever to run an image. The images we work with are 128 x 128, which means that the program's got to loop 268,435,456 (128^4) times just to draw all the lines between pixels! When we first started out, the transform took about $3^1/_2$ hours to run. At present, we've gotten execution time down to just over an hour. I plan to implement a version of the Hough transform in MISP, the Multiple Image Signal Processor; MISP is in C, which should execute faster than DSM's FORTRAN code.

III. The Multiple Image Signal Processor

The Multiple Image Signal Processor, what we like to call MISP, originated in the mind of Starla Christakos, my mentor. Up until this summer, the only signal processing simulation available was DSM, the Digital Seeker Model. DSM, however, works with only one image at a time. This presented a problem when someone wanted to test a set of algorithms on an entire scenario (somewhere in the neighborhood of 1500 images). Our basic goal for MISP, therefore, was to create a signal processing simulation that could process large numbers of images at high speeds.

Derek Holland and I decided (in a fit of brilliant madness) decided that we should program MISP in the C language. This was a brave undertaking, for neither of us had much experience with C and led us to sit staring at computer monitors for days, asking ourselves "How?" Then, one day, everything fell together, and the program seemed to fly into place. We spent the better part of three days writing the program and the better part of three weeks debugging it.

In the end, MISP was a success. We processed 1493 images in 84 minutes--one image about every 3.4 seconds--much faster than anything before its time. There are still some modifications I'd like to make to increase, if not its speed, its modularity. But those will have to wait...

MISP is broken down into five main sections: program control, signal processing, graphics, file input/output, and user interface. Breakdown between sections is as follows:

Program control: main(), quitpr(), and run_simulation();

Signal processing: signal_process(), swap_centroids, and zero_cent();

Graphics: graph_cent(), graph_img(), graph_proc(), graph_set(), thresh_img(), thresh_proc(), and viewimg();

File input/output: init_final(), load_image(), open_script(), read_script(), save_final(), and save_process();

User interface: menu(), menu_display(), and term_set().

PROGRAM CONTROL. Main() and run_simulation() are both simply drivers for the program. Quitpr(), as its name implies, quits the program with an exit status of 0.

SIGNAL PROCESSING. Signal_process() contains all the algorithms for processing the image. Swap_centroids() and zero_cent() perform the dull yet necessary task of cleaning up the centroid arrays between images.

GRAPHICS. The job of setting up the windows and activating the command windows goes to graph_set(). Graph_img() and graph_proc() graph the original image and the processed image, respectively. Thresh_img() and thresh_proc() perform an upper threshold on the display arrays. Viewimg() contains a bit of file I/O, too--it loads an image and displays it. Graph_cent() was supposed to graph the centroids on the image, but that remains to be seen.

FILE INPUT/OUTPUT. Open_script() and read_script() (obviously) open and read the script file. Load_image() loads the image into the intensity array. Save_process() saves the signal processed image to disk. Init_final() and save_final() take care of saving the evaluation of signal processing to disk image by image.

USER INTERFACE. (Again, the obvious:) Menu() and menu_display() are the menu section of the program. Menu() is the "guts" of this section, passing its information on to menu_display(), which displays the current settings of variables. Term_set(), like graph_set(), its counterpart, sets up the program to run on a terminal.

IV. Acknowledgements

Starla Christakos--for everything

Derek Holland--for introducing me to the "world of the HSAP"

Bob "Au Gratin" LeBeau--for providing a unique view of life

Allen Andrews--for the use of his desk & all his help with C

Charles Coker--for his SUN know-how

Matt & TJ--for their advice from "Co-Op Corner"

Dot Sauer--for the chair she'll never know I have

Mike Couvillion--for never-failing computer support

Don Harrison--for heading up this program

All the people in SAI--for taking time out to share their knowledge

Bibliography

Bellman, Richard. Introduction to Matrix Analysis. McGraw-Hill. New York, NY. 1960.

Clancy, Tom. Red Storm Rising. Berkley Books. New York, NY. 1987.

Hansen, Augie. C Programming. Addison-Wesley. Reading, MA. 1989.

Gonzalez, Rafael C. et al. Digital Image Processing. 2 ed. Addison-Wesley. Reading, MA. 1987.

MacDuffee, Cyrus C. Vectors and Matrices. The Mathematical Association of America. Menasha, WI. 1961.

Pratt, William K. Digital Image Processing. John Wiley and Sons. New York, NY. 1978.

Schreier, O. et al. Modern Alegbra and Matrix Theory. Chelsea Publishing Co. New York, NY. 1955.

Sage, Russell G. Tricks of the UNIX Masters. Howard W. Sams and Co. Carmel, IN. 1987.

Soucek, Branko et al. Neural and Massively Parallel Computers. John Wiley and Sons. New York, NY. 1988.

SunOS 4.0 Reference Manual. 13 Volumes. Sun Microsystems, Inc. 1988.

Waite, Mitchell et al. C Primer Plus. Howard W. Sams and Co. Indianapolis, IN. 1987.

The Hough Transform

```
c-----------------------------------------
c-------Hough Transform
c-------TU and DH  6/06/90
c------

        subroutine houghtrans(pfpa,fsize)
        real*4 pfpa(128,128)
        integer fsize, i, j, c, d, max,count
        integer m,b,stb,edb
        real*8 sc
        integer slpsz
        integer slope(-128:129,-12800:12800)
        common / buffers/ buff
        print*,'Hough reached.'
        slpsz = 12800

c-------Determine all possible lines and plot intensities in slope()
        print*,'Determinig lines and plotting in slope()'
        sc = 1.27
        max = 0
        do 2 i = 2,fsize - 1
          do 1 j = 2,fsize - 1
            if (pfpa(i,j).ne.0) then
              do 4 c = 2,fsize - 1
                do 3 d = 2,fsize - 1
                  if (i-c.ne.0) then
                    m = (j - d)/(i - c)
                  else
                    m = fsize + 1
                  endif
                    b = j - (m*i)
                    slope(int(m),int(b/sc)) = slope(int(m),
     *                int(b/sc)) + 1
                    if (slope(int(m),int(b/sc)).gt.max) then
                      max = slope(int(m),int(b/sc))
                    endif
3               continue
4             continue
            endif
1         continue
          print*,'ROW = ',i
2       continue

c-------Threshold slope array
        print*,'Thresholding slope()'
        print*,' MAX =',max
        print*,' THRSH =',max*.75
        count = 0
        do 8 m = -fsize,(fsize+1)
          do 7 b = -slpsz,slpsz
            if (slope(m,b).lt.(max*.75)) THEN
              slope(m,b) = 0
              count = count + 1
            endif
7         continue
8       continue
        print *,'# of entries removed = ',count
        print*,'% of entries removed = ',(count*100)/((4*fsize*slpsz)+1)

c-------Cross-reference data in slope() with pfpa()
        stb = -slpsz
        flag = 0
        do 15 b = -slpsz,slpsz,-1
          do 14 m = -fsize,fsize+1
            if (slope(m,b).ne.0) then
              stb = b
```

```
              endif
   14         continue
              print*,'B(step 1) = ',b
   15       continue
            edb = slpsz
            flag = 0
            do 17 b = slpsz,-slpsz,-1
              do 18 m = -fsize,fsize+1
                if (slope(m,b).ne.0) then
                  edb = b
                endif
   18         continue
            print*,'B(step 2) = ',b
   17       continue
            print*,'Cross-referencing data in slope() with pfpa()'
            do 11 i = 2,fsize-1
              do 9 j = 2,fsize-1
                if (pfpa(i,j).ne.0) then
                  flag = 0
                  do 13 m = -fsize,(fsize+1)
                    do 12 b = stb,edb
                      if (j.eq.((m)*i)+(b*sc)) then
                        flag = 1
                        m = 129
                        b = slpsz
                      endif
   12                 continue
   13               continue
                    if (flag.eq.0) then
                      pfpa(i,j) = 0
                    endif
                  endif
   9            continue
               print*,'ROW = ',i
   11         continue

              end
   c-------------------------------------------------
```

The Multiple Image Signal Processor

Version 1.1

Developed July-August 1990 by HSAPs
Troy Urquhart & Derek Holland

```
misp:
        mv misp.c /home/derek/backup/misp.c
        cat header.text main.c final_eval.c graph_cent.c graph_img.c\
                graph_proc.c graph_set.c init_final.c load_image.c menu.c\
                menu_display.c mispget.c open_script.c quitpr.c read_script.c\
                run_simulation.c save_final.c save_process.c signal_process.c\
                swap_centroids.c term_set.c thresh_img.c thresh_proc.c\
                viewimg.c zero_cent.c > misp.c
        cc -g -o MISP misp.c -lm -lsuntool -lsunwindow -lpixrect
        chmod 4511 MISP
```

```
#define VERSION "1.1"

#include <math.h>
#include <stdio.h>
#include <strings.h>
#include <suntool/sunview.h>
#include <suntool/canvas.h>
#include <suntool/panel.h>

#define IMGEXT  ".img"
#define SGPEXT  ".sgp"
#define INTEXT  ".int"
#define AREEXT  ".are"
#define COMPRESSCMD     "compress "
#define COMPRESSEXT     ".Z"
#define UNCOMPRESSCMD   "uncompress "

#define DEFAULTSCRIPT   "/home/troy/misp/script.file"
#define DEFAULTFINAL    "/home/troy/misp/final"
#define DEFAULTCOMPRES  1
#define DEFAULTGAINOFF  1
#define DEFAULTGAIN     1.25329102
#define DEFAULTOFFSET   -0.128529361
#define DEFAULTSCENEST  0
#define DEFAULTTHRESH   1
#define DEFAULTTHRESHV  4
#define DEFAULTCLUST    1
#define DEFAULTAIMIPT   1
#define DEFAULTHARDDT   0
#define DEFAULTTRACKVC  2
#define DEFAULTAIMPT    1
#define DEFAULTTRUP     50
#define DEFAULTTRLO     1
#define DEFAULTKFOUR    0.9853981634
#define DEFAULTKTHREE   1.078097245

Frame frame, panframe, sigframe;
Canvas canvas, sigwin;
Window panel;
```

```
final_eval()

        extern  char    final_file[50];
        extern  FILE    *final_fl;
        char    chr;

        fputs("~\n",final_fl);
        fclose(final_fl);
        final_fl = fopen(final_file,"r");
        while (chr != '~')
        {
                chr = fgetc(final_fl);
                printf("%c",chr);
        }
        fclose(final_fl);
```

```
graph_img()
{
        extern int      idarr[128][128], idarr[128][128];
        extern char     filename[50];

        register Pixwin *pw;
        register int    i,j,x,y;
        u_char          red[128],
                        blue[128],
                        green[128];


        pw = canvas_pixwin(canvas);
        j = 0;
        for(i=0;i<128;i++)  {
                j += 2;
                if(j==256)
                   j = 255;
                blue[i] = green[i] = red[i] = j;
        }

        pw_setcmsname(pw, "HSAP");
        pw_writebackground(pw,1,1,384,384,NULL);

        for(x=1;x<384;x=x+3)  {
                for(y=1;y<384;y=y+3)  {
                        if((idarr[x/3][y/3]==1))  {
                        pw_put(pw, x+1,y+1,1);
                        }
                        if((idarr[x/3][y/3]>=2)&&(idarr[x/3][y/3]<=6))  {
                        pw_put(pw, x, y, 1);
                        pw_put(pw, x+1,y,1);
                        pw_put(pw, x+1,y+1,1);
                        pw_put(pw,x,y+1,1);
                        pw_put(pw,x+2,y,1);
                        pw_put(pw,x+2,y+2,1);
                        pw_put(pw,x,y+2,1);
                        }
                        if((idarr[x/3][y/3]>=7))  {
                        pw_put(pw, x, y, 1);
                        pw_put(pw, x+1,y,1);
                        pw_put(pw, x+1,y+1,1);
                        pw_put(pw,x,y+1,1);
                        pw_put(pw,x+2,y,1);
                        pw_put(pw,x+2,y+1,1);
                        pw_put(pw,x+2,y+2,1);
                        pw_put(pw,x+1,y+2,1);
                        pw_put(pw,x,y+2,1);
                        }
                }
        }
}
```

```
grapn_proc()
{
        extern int      sarr[128][128], sdarr[128][128];
        extern char     filename[50];

        register Pixwin *sw;
        register int    i,j,x,y;
        u_char          red[128],
                        blue[128],
                        green[128];


        sw = canvas_pixwin(sigwin);

        j = 0;
        for(i=0;i<128;i++)  {
                j += 2;
                if(j==256)
                   j = 255;
                blue[i] = green[i] = red[i] = j;
        }

        pw_setcmsname(sw, "HSAP");
        pw_writebackground(sw,1,1,384,384,NULL);

        for(x=1;x<384;x=x+3)  {
                for(y=1;y<384;y=y+3)  {
                        if((sdarr[x/3][y/3]==1))  {
                        pw_put(sw, x+1,y+1,1);
                        }
                        if((sdarr[x/3][y/3]>=2)&&(sdarr[x/3][y/3]<=6))
                        pw_put(sw, x, y, 1);
                        pw_put(sw, x+1,y,1);
                        pw_put(sw, x+1,y+1,1);
                        pw_put(sw,x,y+1,1);
                        pw_put(sw,x+2,y,1);
                        pw_put(sw,x+2,y+2,1);
                        pw_put(sw,x,y+2,1);
                        }
                        if((sdarr[x/3][y/3]>=7))  {
                        pw_put(sw, x, y, 1);
                        pw_put(sw, x+1,y,1);
                        pw_put(sw, x+1,y+1,1);
                        pw_put(sw,x,y+1,1);
                        pw_put(sw,x+2,y,1);
                        pw_put(sw,x+2,y+1,1);
                        pw_put(sw,x+2,y+2,1);
                        pw_put(sw,x+1,y+2,1);
                        pw_put(sw,x,y+2,1);
                        }
                }
        }
}
```

```
static viewimgj(), quitprj(), run_simulationj();

graph_set(argc, argv)
        int argc;
        char **argv; {
        extern char filename[50];
        extern int display;

        frame = window_create(0, FRAME, FRAME_LABEL, "Current Image",
                WIN_X, 330, WIN_Y, 100, FRAME_ARGS, argc, argv, 0);

        canvas = window_create(frame, CANVAS, WIN_HEIGHT, 384,
                WIN_WIDTH, 384, 0);

        sigframe = window_create(frame, FRAME, WIN_X, 400, WIN_Y, 50,

        panframe = window_create(frame, FRAME, WIN_X, 400,

        sigwin = window_create(sigframe, CANVAS, WIN_HEIGHT, 384,
                                WIN_WIDTH, 384, 0);

        panel = window_create(panframe, PANEL, 0);

        panel_create_item(panel, PANEL_BUTTON,
                PANEL_LABEL_IMAGE, panel_button_image(panel, "Run MISP", 5, 0),
                PANEL_NOTIFY_PROC, run_simulationj, 0);

        panel_create_item(panel, PANEL_BUTTON,
                PANEL_LABEL_IMAGE, panel_button_image(panel,

        panel_create_item(panel, PANEL_BUTTON,
                PANEL_LABEL_IMAGE, panel_button_image(panel, "Quit", 5, 0),
                PANEL_NOTIFY_PROC, quitprj, 0);

        printf("MISP %s\n",VERSION);
        printf("Developed by DH & TU (SAI)\n");

        window_fit(sigframe);
        window_fit(frame);
        window_fit(panel);
        window_fit(panframe);

        window_set(sigwin, NULL, NULL, 0);
        window_set(sigframe, FRAME_LABEL, "SigProc", WIN_SHOW, TRUE, 0);
        window_set(panframe, FRAME_LABEL, "Commands", WIN_SHOW, TRUE, 0);

        window_main_loop(frame);
}

static run_simulationj()
{
        run_simulation();
}

static viewimgj()
{
        viewimg();
}

static quitprj()
{
        quitpr();
}
```

```
init_final()

        extern  char    final_file[50];
        extern  FILE    *final_fl;
        extern  int     trackvc;

        final_fl = fopen(final_file,"w");
        fputs("\nMultiple Image Signal Processing simulation\n\n",final_fl);
        fputs("FINAL EVALUATION:  Tracking by ",final_fl);
        if (trackvc == 1) { fputs("Area ",final_fl); }
        if (trackvc == 2) { fputs("Intensity ",final_fl);
        fputs("Centroid\n\n",final_fl);
```

```
load_image()
{
        extern  char    filename[50];
        extern  int     compres, iarr[128][128];
        extern  float   icen[3][10], acen[3][10];
        char    file[56], cmd[67], byte;
        FILE    *fp;
        int     count, x, y;

        if (compres)
        {
                strcpy(cmd,UNCOMPRESSCMD);
                strcat(cmd,filename);
                strcat(cmd,IMGEXT);
                strcat(cmd,COMPRESSEXT);
                system(cmd);
        }

        strcpy(file,filename);
        strcat(file,IMGEXT);

        fp = fopen(file, "r");
        if (fp == NULL)
        {
                printf("\n%cERROR:  could not open file: \"%s.\"\n\n",7,file);
                icen[2][0] = 2; acen[2][0] = 2;
        }
        else { for (x=0; x<=127; x++) { for (y=0; y<=127; y++)
        {
                iarr[x][y] = 0;
                for (count=0; count<=4; count++)
                {
                        byte = fgetc(fp);
                        if (byte<=58 && byte>=48)
                        {
                                iarr[x][y] = iarr[x][y] * 10 + byte - 48;
                        }
                }
        }}
        fclose(fp);
        }
}
```

```
#include </home/troy/misp/Global.h>

int compres, gainoff, scenest, thresn, thresnv, clust;
int aimpt, harddt, trackvc, ntar;
int trup, trlo;

int     iarr[128][128],         idarr[128][128],
        sarr[128][128],         sdarr[128][128];
int     display;

float gain, offset, icen[3][10],acen[3][10];
float oicen[3][10], oacen[3][10];
float kthree, kfour, variable;

char    filesppd[50],filesgp[50],script_file[50],final_file[50],filename[50];
FILE    *scr_fl, *final_fl;

main(argc, argv)
        int     argc;
        char    **argv;
{
        extern int      display;
        char            str[3];

        while (str[0] != 'y' && str[0] != 'Y' && str[0] != 'n' &&
                str[0] != 'N')
        {
                printf("\nDo you wish to enable graphics? (y/n) ");
                gets(str);
        }
        display = (str[0] == 'y' || str[0] == 'Y') ? 1 : 0;
        if (display)
        {
                graph_set(argc, argv);
        }
        if (!display)
        {
                term_set();
        }
        exit(0);
}
```

```
menu()
{
        extern  char    script_file[50],
                        final_file[50];
        extern  int     compres, gainoff, scenest, thresh,
                        threshv, clust, aimpt, harddt, trackvc, trup,
                        trlo;
        extern  float   gain, offset, kthree, kfour, variable;
        char    str[50],
                inpt[5];
        int     count;

/* set default values */
        strcpy(script_file, DEFAULTSCRIPT);
        strcpy(final_file, DEFAULTFINAL);
        compres = DEFAULTCOMPRES;
        gainoff = DEFAULTGAINOFF;
        gain = DEFAULTGAIN;
        offset = DEFAULTOFFSET;
        scenest = DEFAULTSCENEST;
        thresh = DEFAULTTHRESH;
        threshv = DEFAULTTHRESHV;
        clust = DEFAULTCLUST;
        trackvc = DEFAULTTRACKVC;
        trup = DEFAULTTRUP;
        trlo = DEFAULTTRLO;
        aimpt = DEFAULTAIMPT;
        kthree = DEFAULTKTHREE;
        kfour = DEFAULTKFOUR;
        harddt = DEFAULTHARDDT;

        menu_display();
                while (str[0] != 'R')
                {
                        str[0] = '\0';
                        inpt[0] = '\0';
                        printf("\nEnter 'R' to Run simulation, 'Q' to quit, ");
                        printf("or '?' to redisplay settings.\nMISP>");
                        gets(str);
                        if (str[0] == '?')
                        {
                                menu_display();
                        }
                        if (str[0] == 'A')
                        {
                                printf("Script file: ");
                                gets(script_file);
                        }
                        if (str[0] == 'B')
                        {
                                printf("Final evaluation file: ");
                                gets(final_file);
                        }
                        if (str[0] == 'C')
                        {
                                printf("Gain & offset:\n");
                                printf("\t0) none\n");
                                printf("\t1) Standard SPPD algorithms\n");
                                while (inpt[0] != '0' && inpt[0] != '1')
                                {
                                        gets(inpt);
                                }
                                gainoff = inpt[0] - 48;
                        }
                        if (str[0] == 'D')
                        {
```

```
        printf("Scene stabilization:\n");
        printf("\t0) none\n");
        printf("\t1) Standard SPPD algorithms\n");
        while (inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        scenest = inpt[0] - 48;
}
if (str[0] == 'E')
{
        printf("Threshold:\n");
        printf("\t0) none\n");
        printf("\t1) Fixed value\n");
        printf("\t2) Percentage threshold\n");
        while(inpt[0] != '0' && inpt[0] != '1' &&
                inpt[0] != '2')
        {
                gets(inpt);
        }
        thresh = inpt[0] - 48;
}
if (str[0] == 'F')
{
        printf("Clustering:\n");
        printf("\t0) none\n");
        printf("\t1) Standard SPPD algorithms\n");
        while(inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        clust = inpt[0] - 48;
}
if (str[0] == 'G')
{
        printf("Tracking:\n");
        printf("\t1) by Area centroid\n");
        printf("\t2) by intensity centroid\n");
        while (inpt[0] != '1' && inpt[0] != '2')
        {
                gets(inpt);
        }
        trackvc = inpt[0] - 48;
}
if (str[0] == 'H')
{
        printf("Aimpoint offset:\n");
        printf("\t0) none\n");
        printf("\t1) Standard SPPD algorithms\n");
        while(inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        aimpt = inpt[0] - 48;
}
if (str[0] == 'I')
{
        printf("Hardbody detection:\n");
        printf("\t0) none\n");
        printf("\t1) Standard SPPD algorithms\n");
        while (inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        harddt = inpt[0] - 48;
}
```

```
if (str[0] == '2')
{
        printf("Uncompression/Compression:\n");
        printf("\t0) off\n");
        printf("\t1) Standard\n");
        while (inpt[0] != '0' && inpt[0] != '1')
        {
                gets(inpt);
        }
        compres = inpt[0] - 48;
}
if (str[0] == 'z')
{
        return;
}
if (str[0] == 'a')
{
        printf("Gain = ");
        mispget();
        gain = variable;
}
if (str[0] == 'b')
{
        printf("Offset = ");
        mispget();
        offset = variable;
}
if (str[0] == 'c')
{
        printf("Threshold = ");
        mispget();
        threshv = variable;
}
if (str[0] == 'd')
{
        printf("Max. target movement/frame = ");
        mispget();
        trup = variable;
}
if (str[0] == 'e')
{
        printf("Min. target movement/frame = ");
        mispget();
        trlo = variable;
}
if (str[0] == 'f')
{
        printf("K3 constant = ");
        mispget();
        kthree = variable;
}
if (str[0] == 'g')
{
        printf("K4 constant = ");
        mispget();
        kthree = variable;
}
    }
}
```

```
menu_display()
{
        extern  char    script_file[50],
                        final_file[50];
        extern  int     gainoff, scenest, thresh,
                        threshv, clust, aimpt, haradt, trackvc, trup,
                        trlo;
        extern  float   gain, offset, kthree, kfour;

        printf("\n\nMISP setup:\n\n");
        printf("A) Script file:\t\t\"%s\"\n",script_file);
        printf("B) Final file:\t\t\"%s\"\n",final_file);
        printf("C) Gain & offset:\t");
        if (!gainoff)
        {
                printf("none\n");
        }
        if (gainoff == 1)
        {
                printf("Standard SPPD algorithms\n");
                printf("    a) Gain = %f\n    b) Offset = %f\n",gain,offset);
        }
        printf("D) Scene stabilization:\t");
        if (!scenest)
        {
                printf("none\n");
        }
        if (scenest == 1)
        {
                printf("Standard SPPD algorithms\n");
        }
        printf("E) Threshold:\t\t");
        if (!thresh)
        {
                printf("none\n");
        }
        if (thresh == 1)
        {
                printf("Fixed value\n");
        }
        if (thresh == 2)
        {
                printf("Percentage threshold\n");
        }
        if (thresh == 1 || thresh == 2)
        {
                printf("    c) Threshold = %d\n",threshv);
        }
        printf("F) Clustering:\t\t");
        if (!clust)
        {
                printf("none\n");
        }
        if (clust == 1)
        {
                printf("Standard SPPD algorithms\n");
        }
        printf("G) Tracking:\t\t");
        if (trackvc == 1)
        {
                printf("by Area centroid\n");
        }
        if (trackvc == 2)
        {
                printf("by Intensity centroid\n");
        }
```

```
printf("   d) Max. target movement/frame = %d\n",trup);
printf("   e) Min. target movement/frame = %d\n",trlo);
printf("H) Aimpoint offset:\t");
if (!aimpt)
{
        printf("none\n");
}
if (aimpt == 1)
{
        printf("Standard SPPD algorithms\n");
        printf("   f) K3 constant = %f\n",kthree);
        printf("   g) K4 constant = %f\n",kfour);
}
printf("I) Hardbody detection:\t");
if (!harddt)
{
        printf("none\n");
}
if (harddt == 1)
{
        printf("Standard SPPD algorithms\n");
}
printf("Z) Compression:\t\t");
if (!compres)
{
        printf("off\n");
}
if (compres == 1)
{
        printf("Standard\n");
}
```

```
mispget()
{
        extern  float   variable;
        char    inpt[50];
        int     x, y, z, sign, j = 0;
        float   d = 0;
        gets(inpt);
        variable = 0;
        d = 0;
        sign = (inpt[0] == '-') ? -1 : 1;
        x = (sign == 1) ? 0 : 1;
        while (inpt[x] != '.' && x <= strlen(inpt))
        {
                if (inpt[x] <= 57 && inpt[x] >= 48)
                {
                        j = (j*10) + inpt[x] - 48;
                }
                x++;
        }
        if (x < strlen(inpt))
        {
                for (y=x; y<=strlen(inpt); y++)
                {
                        if (inpt[x] <= 57 && inpt[x] >= 48)
                        {
                                d = (d*10) + inpt[x] - 48;
                        }
                }
        }
        for (z=x; z<=y; z++)
        {
                d = d / 10;
        }
        variable = sign * (j + (d));
}
```

```
open_script()

        extern  char    script_file[50],        filename[50];
        extern  FILE    *scr_fl;
        scr_fl = fopen(script_file, "r");
        if (scr_fl == NULL)
        {
                printf("ERROR: Could not open file: %s\n",script_file);
                printf("--MISP aborted\n");
                exit(9);
        }
        read_script();
}
```

```
static quitpr()
{
        window_destroy(frame);
        printf("MISP exit.");
}
```

```
read_script()
{
        extern  char    filename[50];
        extern  FILE    *scr_fl;
        int     count=0;

        while (filename[count-1] != '~' && count <= 50)
        {
                filename[count] = fgetc(scr_fl);
                count++;
        }
        filename[count-1] = '\0';
        count = fgetc(scr_fl);
}
```

```
static run_simulation()
{
        extern int      display;

/* variables for file i/o, etc. */
        extern  char    filename[50];

        extern  FILE    *scr_fl, *final_fl;
        extern  char    script_file[50],
                        final_file[50];
        extern  float   icen[3][10];
        menu();
        if (script_file[0] == '\0') { strcpy(script_file,DEFAULTSCRIPT); }
        if (final_file[0] == '\0') { strcpy(final_file,DEFAULTFINAL); }
        open_script();
        init_final();
        while(filename[0] != '\0')
        {
                zero_cent();
                load_image();
                if(icen[2][0] != 2)
                {
                        if(display==1)
                        {
                                thresh_img();
                                graph_img();
                        }
                        signal_process();
                        if (display==1)
                        {
                                thresh_proc();
                                graph_proc();
                        }
                        save_process();
                }
                save_final();
                swap_centroids();
                read_script();
                printf("\n");
        }
        close(scr_fl);
        final_eval();
        script_file[0] = '\0';
        printf("\nMISP EXECUTE COMPLETE.\n\n");
}
```

```
save_final()
{
        extern  char    filename[50];
        extern  FILE    *final_fl;
        extern  float   icen[3][10],    acen[3][10];
        extern  int     trackvc;
        int     cen[3][10],
                x, y;

        for (x=0; x<=2; x++) { for (y=0; y<=9; y++)
        {
                if (trackvc == 1) { cen[x][y] = acen[x][y]; }
                if (trackvc == 2) { cen[x][y] = icen[x][y]; }
        }}
        fputs(filename,final_fl);
        fputs(":\n",final_fl);
        if (cen[2][0] == 2) { fputs("\t\t\tFILE LOAD ERROR\n",final_fl); }
                else {
        for (y=0; y<=9; y++) { if (cen[0][y] != 129) { for (x=0; x<=1; x++)
        {
                if (x==0) { fputs("  Target ",final_fl);
                        fputc(y+48,final_fl);fputs(" == (",final_fl); }
                if (cen[x][y] < 10)
                {
                        fputc(cen[x][y]+48,final_fl);
                }
                if (cen[x][y] >=10 && cen[x][y] < 100)
                {
                        fputc(cen[x][y]/10+48,final_fl);
                        fputc(cen[x][y]%10+48,final_fl);
                }
                if (cen[x][y] >=100)
                {
                        fputc(cen[x][y]/100+48,final_fl);
                        fputc(cen[x][y]/10%10+48,final_fl);
                        fputc((cen[x][y]%100)+48,final_fl);
                }
                if (x==0) { fputs(",\t",final_fl); }
                if (x==1) { fputs(")\t:: ",final_fl); }
                } if (cen[2][y] != 1) { fputs("UN",final_fl); }
                fputs("Successful\n",final_fl);
        }}}
        fputc('\n',final_fl);
}
```

```
save_process()
{
        extern  char    filename[50];
        extern  int     sarr[128][128];
        char    group[5], name[54], cmd[63];
        FILE    *fp;
        int     x,y,z;

        strcpy(name, filename);
        strcat(name, SGPEXT);

        fp = fopen(name, "w");

        for (x=0; x<=127; x++) { for (y=0; y<=127; y++)
        {
                if (sarr[x][y] < 10)
                {
                        fputs("    ",fp);
                        fputc((sarr[x][y] + 48),fp);
                }
                if (sarr[x][y] >= 10 && sarr[x][y] < 100)
                {
                        for (z=0; z<=2; z++) { fputc(' ',fp); }
                        fputc(((sarr[x][y] / 10) + 48),fp);
                        fputc(((sarr[x][y] % 10) + 48),fp);
                }
                if (sarr[x][y] >= 100 && sarr[x][y] < 1000)
                {
                        for (z=0; z<=1; z++) { fputc(' ',fp); }
                        fputc(((sarr[x][y] / 100) + 48),fp);
                        fputc((((sarr[x][y] % 100) / 10) + 48),fp);
                        fputc(((sarr[x][y] % 10) + 48),fp);
                }
                if (sarr[x][y] >= 1000 && sarr[x][y] < 10000)
                {
                        fputc(' ',fp);
                        fputc(((sarr[x][y] / 1000) + 48),fp);
                        fputc((((sarr[x][y] % 1000) / 100) + 48),fp);
                        fputc((((sarr[x][y] % 100) / 10) + 48),fp);
                        fputc(((sarr[x][y] % 10) + 48),fp);
                }
                if (sarr[x][y] >= 10000 && sarr[x][y] <100000)
                {
                        fputc(((sarr[x][y] / 10000) + 48),fp);
                        fputc((((sarr[x][y] % 10000) / 1000) + 48),fp);
                        fputc((((sarr[x][y] % 1000) / 100) + 48),fp);
                        fputc((((sarr[x][y] % 100) / 10) + 48),fp);
                        fputc(((sarr[x][y] % 10) + 48),fp);
                }
        } fputc('\n',fp); }
        fclose(fp);

        if (compres == 1)
        {
                strcpy(cmd,COMPRESSCMD);
                strcat(cmd,name);
                system(cmd);
                strcpy(cmd,COMPRESSCMD);
                strcat(cmd,filename);
                strcat(cmd,IMGEXT);
                system(cmd);
        }
}
```

```
signal_process()

        extern int gainoff, scenest, thresh, thresnv, clust;
        extern int trup, trlo, ntar;
        extern int iarr[128][128], sarr[128][128];
        extern float gain, offset, icen[3][10],acen[3][10];
        extern float kthree, kfour;

        extern char filename[50];

        int target[128][128], i, n, row, col, g, h, avel, ivel, j, y, x;
        int a, b, c, d, e, f;
        float  valux, valuy, valu;
        float xca, yca, xsm, ysm, xv, yv, acm;
        float rowmoa, colmoa, rowmom, colmom, rowtoa, rowtot;
        float total, totala, maxtwo, yaim, xaim;

        printf("\nSignal Processing image %s.\n\n", filename);
        for(x=0;x<128;x++)  {
                for(y=0;y<128;y++)  {
                        sarr[x][y] = iarr[x][y];
                }
        }

/*      Gain and Offset section.  */

        if(gainoff == 1)  {
                printf("Running SPPD Gain and Offset\n");
                for(x=0;x<128;x++)  {
                        for(y=0;y<128;y++)
                                sarr[x][y] = (gain*sarr[x][y]) + offset;
                }
        }

        if(gainoff == 0)  {
                /*printf("No Gain and Offset selected\n");*/
        }

/*      Scene stabilization section.  */

        if(scenest == 0) {
                /*printf("No Scene Stabilization selected\n");*/
        }

/*      Threshold Section.  */

        if(thresh == 1)   {
                printf("Threshold = %d\n",threshv);
                for(x=0;x<128;x++)  {
                        for(y=0;y<128;y++)  {
                                if(sarr[x][y] < threshv)
                                        sarr[x][y] = 0;
                        }
                }

        }

        if(thresh == 0)  {
                /*printf("No threshold selected\n");*/
        }

/*      Clustering section      */

        if(clust == 1)  {
                ntar = 0;
                for(x=0;x<128;x++)  {
```

```
                        for(y=0;y<128;y++)  {
                                if(sarr[x][y] > 0)  {
                                        if(ntar == 0) {
                                                target[x][y] = 1;
                                                ntar = 1;
                                        }
                                        else if(target[x][y-1] > 0) {
                                           target[x][y] = target[x][y-1];
                                        }
                                        else if(target[x-1][y-1] > 0) {
                                           target[x][y] = target[x-1][y-1];
                                        }
                                        else if(target[x-1][y] > 0) {
                                           target[x][y] = target[x-1][y];
                                        }
                                        else if(target[x-1][y+1] > 0) {
                                           target[x][y] = target[x-1][y+1];
                                        }
                                        else {
                                           ntar = ntar + 1;
                                           target[x][y] = ntar;
                                        }
                                }
                        }
                }
                printf("Number of Targets: %d\n", ntar);
        }

        if(clust == 0)  {
                /*printf("No clustering selected\n");*/
        }

/*      Centroid Area.  Figures Area and Intensity      */

        if(ntar > 10)ntar = 10;

        n = 0;
        j = 0;

        for(i=1;i<=ntar;i++)  {
                rowmoa = 0.0;
                colmoa = 0.0;
                rowmom = 0.0;
                colmom = 0.0;
                total = 0.0;
                totala = 0.0;
                j = 0;

                for(row=0;row<128;row++)  {
                        rowtoa = 0.0;
                        rowtot = 0.0;
                        for(col=0;col<128;col++)  {
                                if((target[row][col]) == i)  {
                                        j+=1;
                                        colmom = colmom+(sarr[row][col]*
                                        rowtot = rowtot + sarr[row][col];
                                        colmoa = colmoa + (1.0*(col-.5));
                                        rowtoa = rowtoa + 1.0;
                                }
                        }
                        total = total + rowtot;
                        rowmom = rowmom + (rowtot*(row - .5));
                        totala = totala + rowtoa;
                        rowmoa = rowmoa + (rowtoa*(row-.5));
                }
```

```
                if(j > 20)  {
                        n+=1;
                        icen[1][n-1] = rowmom/total;
                        icen[0][n-1] = colmom/total;
                        acen[1][n-1] = rowmoa/totala;
                        acen[0][n-1] = colmoa/totala;

                        printf("\nTARGET :%d\n", n);
                        printf("Intensity Centroid = %f , %f\n",icen[0][n-1],
                                        icen[1][n-1]);
                        printf("Area Centroid = %f , %f\n",acen[0][n-i],
                                        acen[1][n-1]);
                        printf("Pixels in target = %d\n", j);
                }
        }
        printf("\n%d target(s) less than 20 pixels.\n\n", (ntar-n));
        ntar=n;


/*      Track Area      */

        if(trackvc == 1)  {
                        for(i=0;i<ntar;i++)  {
                                avel = sqrt(((acen[0][i]-oacen[0][i])*(acen[0][i]-c
                                printf("PIXELS MOVED = %d\n", avel);
                                if((avel <= trup)&&(avel >= trlo))  {
                                        acen[2][i] = 1;
                                        printf("TARGET : %d\n",i+1);
                                        printf("-- Track: Successful.\n");
                                }
                                else    {
                                        acen[2][i] = 0;
                                        printf("TARGET : %d\n",i+1);
                                        printf("-- Track: Failed.\n");
                                }
                        }
        }

        if(trackvc == 2)
                        f   i=0;i<ntar;i++)  {
                                ivel = sqrt(((icen[0][i]-oicen[0][i])*(icen[0][i]-c
                                printf("PIXELS MOVED = %d\n", ivel);
                                if((ivel <= trup)&&(ivel >= trlo))  {
                                        icen[2][i] = 1;
                                        printf("TARGET : %d\n",i+1);
                                        printf("-- Track : Successful.\n");
                                }
                                else    {
                                        icen[2][i] = 0;
                                        printf("TARGET : %d\n",i+1);
                                        printf("-- Track: Failed.\n");
                                }
                        }
        }

/*      Aimpoint-offset         */

        if(aimpt == 1) {
                for(n=0;n<ntar;n++)  {
                        if((acen[2][n]==1)||(icen[2][n]==1))
                                for(i=0;i<128;i++)  {
                                        for(j=0;j<128;j++)  {
                                                if(sarr[i][j]>maxtwo) {
                                                        maxtwo = sarr[i][j];
                                                        x = i;
```

```
                                        y = j;
                                    }
                                    if(sarr[i][j]>0) {
                                        a = a + h;
                                        b = b + 1;
                                        c = c + g;
                                        d = d + h*h;
                                        e = e + g*g;
                                        f = f + g*h;
                                    }
                                }
                            }
                            xca = a/b;
                            yca = c/b;
                            xsm = d/b;
                            ysm = e/b;
                            xv = xsm - xca*xca;
                            yv = ysm - yca*yca;
                            acm = f/b;
                            if(((fabs(acen[0][n] - icen[0][n]))<=(kthree*

                                xaim = acen[0][n];
                                yaim = acen[1][n];
                            }
                            else    {

                                xaim = kfour*(icen[0][n] - fabs(icen
                                yaim = kfour*(icen[1][n] - fabs(icen
                            }
                            printf("TARGET : %d\n", n+1);
                            printf("Aimpoint offset :\n");
                            printf("===>%f , %f\n", xaim, yaim);
                        }
                        else    {
                            printf("TARGET : %d\n", n+1);
                            printf("Warning:: Tracking FAILED.\n");
                            printf("--Necessitates Offset CANCEL.\n");
                        }
                    }
                }

                if(aimpt ==0)   {
                    /*printf("No Aimpoint Offset selected.\n");*/
                }

/*      Hardbody Detection Section (EXP)        */

                if(harddt == 0)   {
                    /*printf("Hardbody detect currently inactive.\n");*/
                }

}
```

```
swap_centroids()
{
        extern  float   icen[3][10],    oicen[3][10],
                        acen[3][10],    oacen[3][10];
        int     x, y;

        for (x=0; x<=2; x++) { for (y=0; y<=9; y++)
        {
                oicen[x][y] = icen[x][y];
                oacen[x][y] = acen[x][y];
        }}
}
```

```
term_set()
{
        extern int display;

        printf("\n\nMISP %s\n",VERSION);
        printf("Developed by DH & TU (SAI)\n");
        run_simulation();
}
```

```
thresh_img()

        extern  int     iarr[128][128], idarr[128][128];
        int     x, y;

        for (x=0; x<=127; x++) { for (y=0; y<=127; y++)
        {
                idarr[x][y] = (iarr[x][y] <= 128) ? iarr[x][y] : 128;
        }}
```

```
thresh_proc()
{
        extern  int     sarr[128][128], sdarr[128][128];
        int     x, y;

        for (x=0; x<=127; x++) { for (y=0; y<=127; y++)
        {
                        sdarr[x][y] = (sarr[x][y] <= 128) ? sarr[x][y] : 128;
        }}
}
```

```
static viewing()

        char    byte;
        extern  char    filesgp[50],filesppc[50];
        FILE    *fp;
        int     count, cx, cy, suffix, m;
        extern int iarr [128][128], idarr[128][128];
        char suf[3];


        while(suf[0] != 'A' && suf[0] != 'B' && suf[0] != 'C')
        {
                printf("\nEnter type of image:\n");
                printf("\tA)  Initial SPPD.\n");
                printf("\tB)  Signal Processed SPPD.\n");
                printf("\tC)  Automatic selection.\n");
                printf("\nVIEW>");
                gets(suf);
        }

        suffix = suf[0] - 64;

        printf("image: ");
        gets(filesppd);
        if (filesppd[0] == '\0') {
                fprintf(stderr, "ERROR: no filename typed.\n");
                exit(0);
        }

        strcpy(filesgp,filesppd);
        strcat(filesppd, IMGEXT);
        strcat(filesgp, SGPEXT);

        for (m=0; m<=1; m++)
        {
        if (suffix == 1 || (suffix == 3 && m==0))
        {
                fp = fopen(filesppd, "r");
        }
        if (suffix == 2 || (suffix == 3 && m==1))
        {
                fp = fopen(filesgp, "r");
        }
        if (fp != NULL)
        {
                for (cx=0; cx<=127; cx++) { for (cy=0; cy<=127; cy++)
                {
                        iarr[cx][cy] = 0;
                        for (count=0; count<=4; count++) {
                                byte = fgetc(fp);
                                if (byte>=48 && byte<=57) {
                                        iarr[cx][cy] = iarr[cx][cy] * 10 +
                                                        byte - 48;
                                }
                        }
                }}
                fclose(fp);

                printf("Scaling intensity array for display . . . ");
                for (cy=0; cy<=127; cy++) { for (cx=0; cx<=127; cx++)
                {
                        idarr[cx][cy] = (iarr[cx][cy] <= 128)?iarr[cx][cy]:128;
                }}
                dispimg(suffix, m);
                if (suffix == 1 || suffix == 2) { m = 2; }
         }}
```

```
}

static dispimg(suffix,m)
        int suffix, m;
{
        extern int      iarr[128][128], idarr[128][128];
        extern char     filename[50];

        register Pixwin *pw;
        register int    i,j,x,y;
        u_char          red[128],
                        blue[128],
                        green[128];

        if(suffix==1)pw = canvas_pixwin(canvas);
        if(suffix==2)pw = canvas_pixwin(sigwin);
        if(suffix==3 && m==0)pw = canvas_pixwin(canvas);
        if(suffix==3 && m==1)pw = canvas_pixwin(sigwin);

        j = 0;
        for(i=0;i<128;i++)  {
                j += 2;
                if(j==256)
                   j = 255;
                blue[i] = green[i] = red[i] = j;
        }

        pw_setcmsname(pw, "HSAP");
        pw_writebackground(pw,1,1,384,384,NULL);

        for(x=1;x<384;x=x+3)  {
                for(y=1;y<384;y=y+3)  {
                        if((idarr[x/3][y/3]==1))  {
                        pw_put(pw, x+1,y+1,1);
                        }
                        if((idarr[x/3][y/3]>=2)&&(idarr[x/3][y/3]<=6))  {
                        pw_put(pw, x, y, 1);
                        pw_put(pw, x+1,y,1);
                        pw_put(pw, x+1,y+1,1);
                        pw_put(pw,x,y+1,1);
                        pw_put(pw,x+2,y,1);
                        pw_put(pw,x+2,y+2,1);
                        pw_put(pw,x,y+2,1);
                        }
                        if((idarr[x/3][y/3]>=7))  {
                        pw_put(pw, x, y, 1);
                        pw_put(pw, x+1,y,1);
                        pw_put(pw, x+1,y+1,1);
                        pw_put(pw,x,y+1,1);
                        pw_put(pw,x+2,y,1);
                        pw_put(pw,x+2,y+1,1);
                        pw_put(pw,x+2,y+2,1);
                        pw_put(pw,x+1,y+2,1);
                        pw_put(pw,x,y+2,1);
                        }
                }
        }
        graph_cent();
}
```

```
zero_cent()
{
        extern  float   icen[3][10],
                        acen[3][10];
        int     x, y;

        for (x=0; x<=2; x++) { for (y=0; y<=9; y++)
        {
                icen[x][y] = 129;
                acen[x][y] = 129;
        }}
}
```

Ballistics Applications in Aerospace Research

AFATL/FXA

Greg VanWiggeren

Mentor:
Capt. Mike Valentino

August 15, 1990

Table of Contents

I. Introduction

II. Acknowledgements

III. A Description of My Research

IV. Conclusion

V. Experiences Gained

I. Introduction...

This summer was a time I will not forget. I learned a great deal about engineering as a career and gained many valuable experiences. I expected to learn this summer, but I did not expect to learn so much. My work at the aeroballistics branch was highly educational. I learned not only about my own field of study, aeroballistics, but also about other technical fields, and engineering in general. I visited other apprentices from other branches of AFATL to learn about their work and see the different areas of research. I have always been fascinated by the science of flight and this summer was my first opportunity to pursue it. I really appreciated the opportunity to learn hands-on and work in a real-world environment.

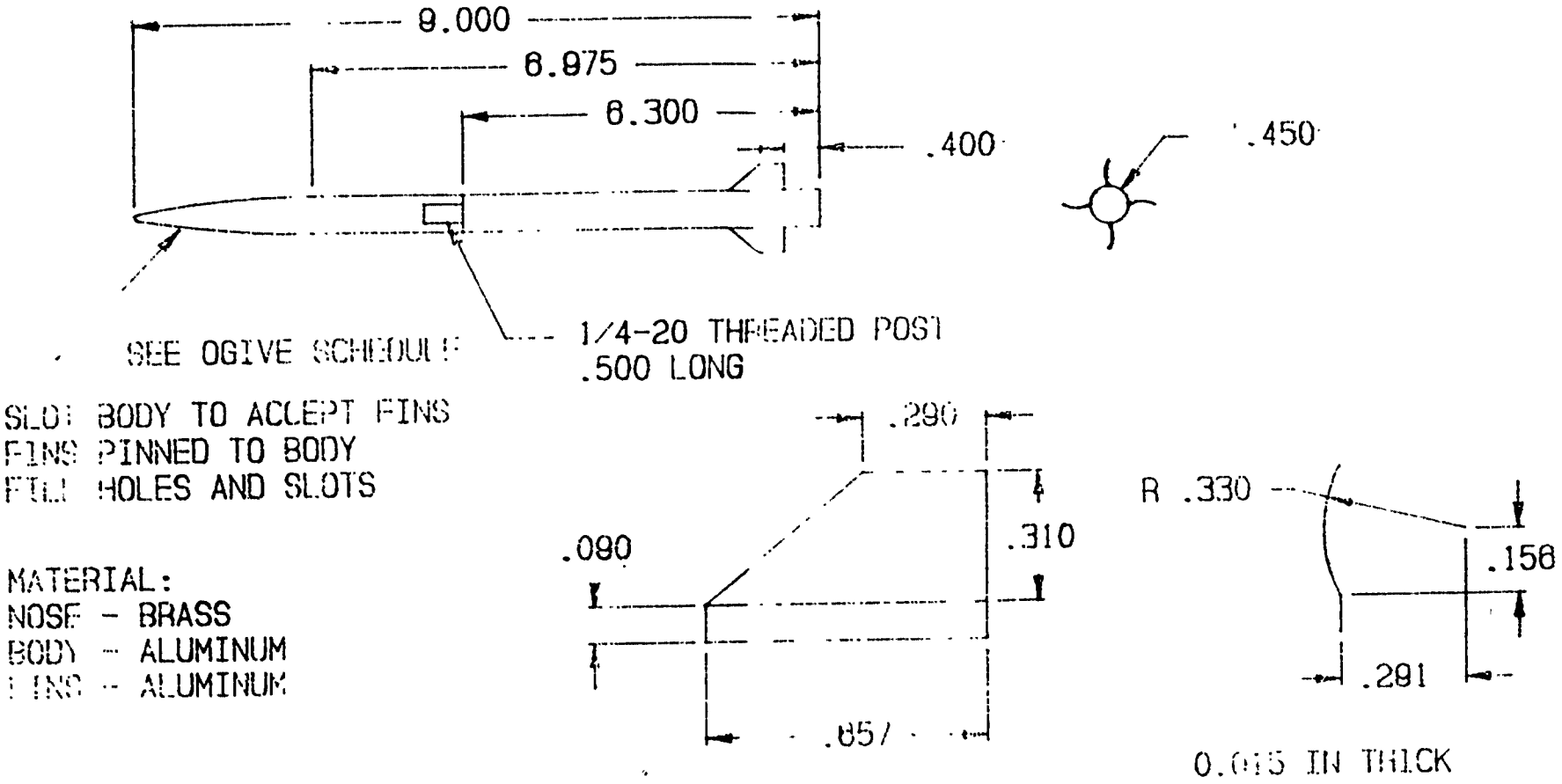
II. Acknowledgements...

I would like to thank the people who allowed me the opportunity to participate in the program. Specifically, I would like to thank my mentor Capt. Mike Valentino for his guidance and technical expertise. I appreciate the extra effort that Capt. Valentino expended in my behalf. I would also like to thank Don Harrison, the program director, for his efforts. Special thanks to Lieutenant Mike Stevens for laughing at my jokes, Karl Pitsch for his help with the German language, and Bernd Becker for keeping my car running. I also want to thank Lieutenant Rico Vitale for his philosophy lessons, and Greg Abate for teaching me all the words to "Born to Run".

III. A Description of My Research...

Aeroballistics is a field that is primarily concerned with missiles, rockets, bullets, and other projectiles. We test and research various projectile designs, usually missile configurations or improved bullets, but sometimes even subscale aircaraft. Our projects come from three sources: outside contractors, other DoD organizations, and in-house research. The improved bullets are usually from twenty to forty millimeters in diameter and are designed to maintain higher speeds for longer distances. The bullets we test may someday be used by various aircraft in the Air Force inventory. The missile configurations are usually designed around a unique purpose. For example, we have tested ground-to-ground missiles for the army, anti-aircraft missiles for the Air Force, and even some in-house missiles with special fins to allow them to be shot from a cylindrical launch tube. The missile that I worked on involved just such a special fin design. (Fig.1)

Our building, 415, is rather small; we only have eight engineers in our section. Three are Air Force active duty, three are civil service, and two are exchange engineers from Germany. The aeroballistics section of FXA uses two ranges: the ARF, Aeroballistics Research Facility, and the BEF, Ballistics Experimentation Facility. Most of our detailed testing and research takes place at the ARF. The ARF is a building 207 meters long with a cross-sectional area of approximately four square meters for the first sixty-nine meters and five square meters for the last 138. 131 camera stations, of which about fifty-five are utilized at the present time, are placed at intervals along the range. Each station has two cameras, two reflective screens, two spark stations, and an infra-red detector. The cameras and reflectors are placed opposite one



AIR FORCE NVM -- BURN OUT CG, 8 MODELS

Fig.1

another. The projectile is shot from one of several guns at the end of the range. As it flies down range, it passes in front of a camera station and is detected by the infra-red detector. This in turn ignites the sparks which expose the film in the cameras as the missile flies past. The cameras do not actually photograph the missile; they photograph the missile's shadow on the reflective screens.

The ARF is called a six degree of freedom ballistics range because, from the photographs, data can be obtained that describes the missile's movement on the three axes - X,Y,Z. X is the axis that extends from one end of the range to the other. Y is the left-right axis, and Z is the up-down axis. The pictures also give us data about the rotation of the projectile about the X,Y, and Z axes - roll, pitch, and yaw respectively. These types of movement are the six degrees of freedom. Because we can obtain data about all six degrees of freedom, we have some advantages over traditional windtunnel testing which allows only limited movement.

The ARF is only the first step in the interactive data analysis I completed this summer. (Fig.2) The majority of the time I spent working on my summer project, I worked with the ARFDAS, Aeroballistics Research Facility Data Analysis System. ARFDAS is actually a group of programs, of which I worked mostly with the Linear Theory. The objective of Linear Theory is to provide the researchers with data about the projectile's motion which might be helpful in designing the full-scale model.

Before the analysis of the projectile can take place, the computer must know some of the physical properties of the missile and some of the air properties. The list of physical properties includes the following: Diameter, weight, axial inertia, transverse [Y] inertia, transverse [Z] inertia, length, and center of gravity location. The diameter is used to

EXPERIMENTAL AERODYNAMICS & BALLISTICS
DATA ANALYSIS SYSTEM
AEROBALLISTIC RESEARCH FACILITY

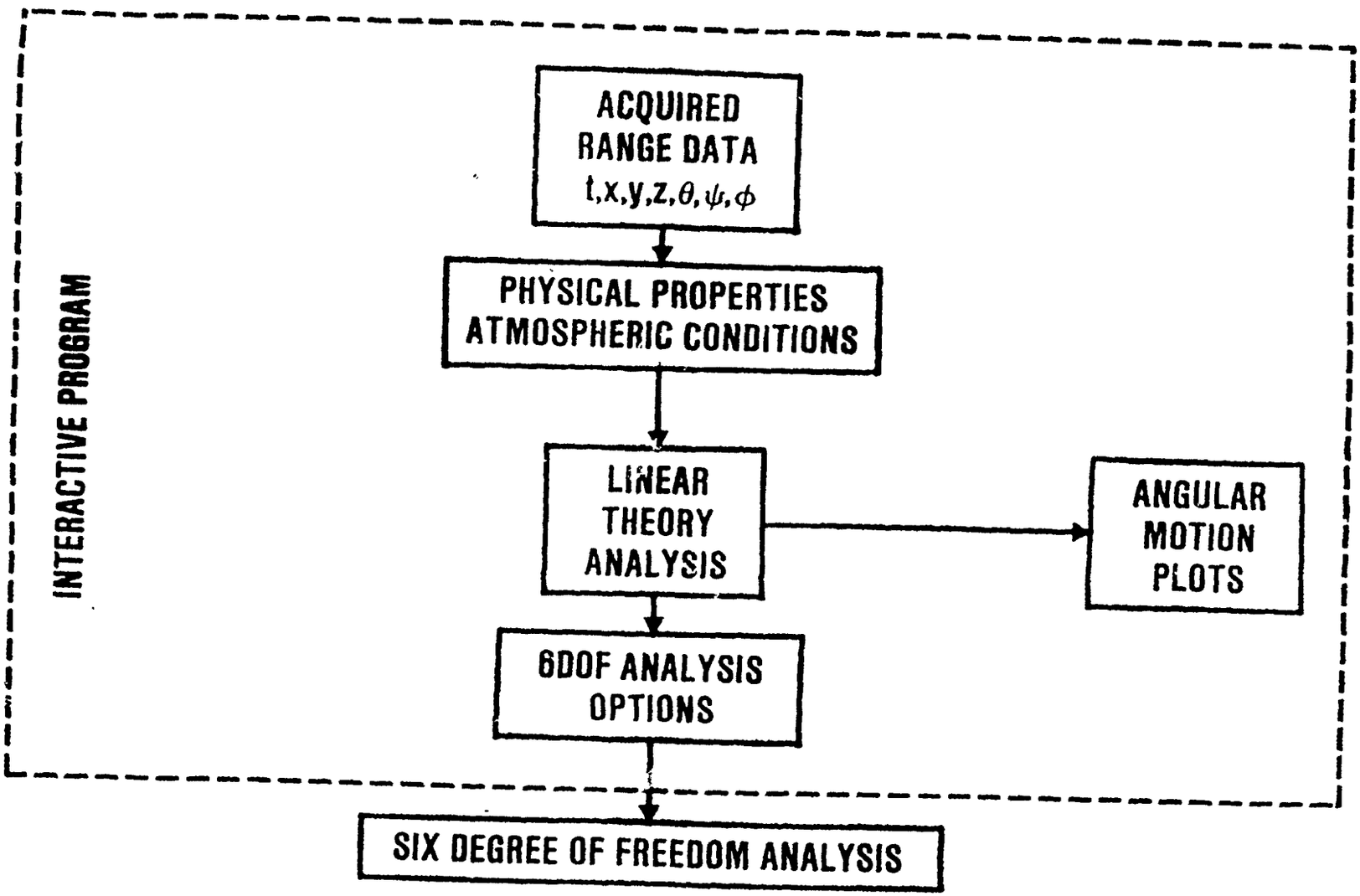
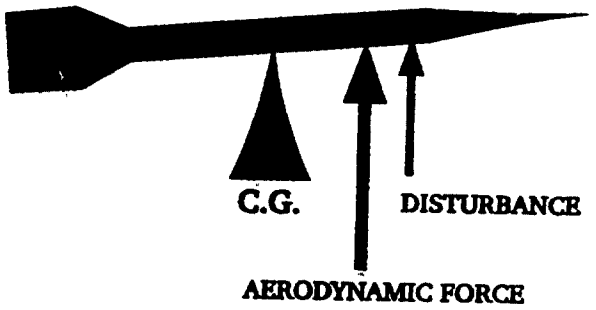


Fig.2

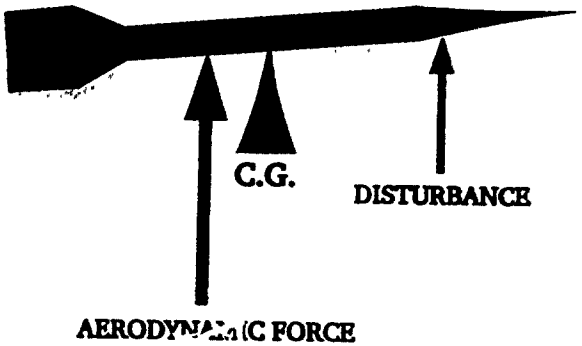
calculate several aerodynamic coefficients. The axial inertia is the resistance of the missile to rotation around its axis. The transverse [Y] inertia is the resistance of the missile to rotation around the Y axis - pitch. In an axisymmetric missile, the value for the transverse [Z] inertia equals the value of the transverse [Y] inertia. Length, like diameter, is used in calculations of aerodynamic coefficients. The location of the center of gravity represents a key determining factor in the computer's predictions of stability. The center of pressure must be behind the center of gravity for the projectile to be statically stable (fig.3), but the computer must know some of the conditions of the air to more accurately predict the center of pressure.The air properties necessary for program calculations include the following: density, speed of sound, temperature, pressure, and relative humidity. Once the computer has all the input it needs, the analysis can begin.

Linear Theory uses equations that describe the path of an object trough a fluid, such as air, to calculate a theoretical trajectory for the missile. After I input some preliminary values for coefficients of these equations the program calculates a trajectory. I then compare the actual data to the theoretical data and through an iterative process, adjust the values of the coefficients of the equations until the program's theoretical trajectory matches the actual trajectory. This process is called 'fitting'. I started on this project just a few days after arriving, and I should be finished by the end of this week.

When the analysis is run, a screen is displayed on the tektronix 4107 terminal that allows you to adjust some of the vector properties of the projectile's flight, specifically the precession and nutation vectors. By adjusting the vectors, the coefficients of the linear theory equations are adjusted. The properties include the vector's frequency,



UNSTABLE Cma



STABLE Cma

Fig.3

orientation, amplitude, damping, frequency change, and trim. At first, the vector properties must be adjusted manually, but once the theoretical equations begin to resemble the actual trajectory, the program can be set free to fine tune the coefficient values. Each 'shot' (projectile) is different, and it takes a knowledge of aerodynamics and practice with the linear theory program to correctly reduce a 'shot'.

A great deal of knowledge is needed to actually analyze a shot. It has taken me all summer to be able to interpret even part of the analysis. Each coefficient reveals a certain facet of that missile's overall aerodynamics, but the coefficients, when used in combination, paint a complete picture of its overall flight characteristics. To be honest, most of the engineers can't remember the importance of every coefficient every day. The paçes of attachment one are typical Linear Theory output.

IV. Conclusion...

Of course, the goal of this analysis is to predict the aerodynamic properties of a full-scale model. After all, who wants a half-scale model? For my summer project, I analyzed data from twenty-four 'shots' of a generic high length to diameter ratio missile with a wrap-around fin configuration. Each missile was shot at a different speed from our thirty millimeter two stage light-gas-gun. Unfortunately, the research was cancelled, and as a result, it will probably never become a full-scale model. However, some conclusions can be drawn from our experiments. For example, the wrap-around fins cause a side-moment, and we are able to determine the mach at which the missile becomes statically unstable. Although the missile is not likely to become a full-scale model, the data we obtained can be built upon or used in the future.

V. Experiences Gained...

Before spending money to build a model and more money to test it, we try to predict some of its aerodynamic properties. I worked with two preliminary design programs this year; one was called Prodas, and the other was called Missile Datcom. In Prodas, the user inputs the missile's configuration according to the program's format; for example, the user tells the program how long the centerbody is, describes the fin shape, describes the nose shape, inputs the center of gravity location, and much more. Prodas then searches its extensive library for another similar missile, and from that missile's aerodynamic properties, it is possible to gain an insight into the new missile's properties. Attachment two represents some typical Prodas output.

Missile Datcom uses similar techniques to input the missile configuration to be tested, but it does not have a library of other previously constructed missiles to predict the properties of the new design. Instead, it uses a variety of mathematical techniques to calculate the properties of the missile. Both Prodas and Missile Datcom have advantages, but I prefer Missile Datcom over Prodas. Its input methods are easier to use; however, only Prodas allows you to see the geometry of the missile as it is being constructed. Attachment three is typical output from Missile Datcom.

Missile Datcom is a relatively recent addition to our preliminary design program repertoire. Capt. Valentino asked me to learn to use the program and handed me the manual. He wanted me to compare output from Missile Datcom to Prodas and the empirical data obtained form linear theory. After a summer of manual reading and testing, my knowledge of the program is still incomplete, but I know more about modeling missiles

on Missile Datcom than anyone in the section.

Captain Valentino also wanted to be able to use the plot output from Missile Datcom to make graphs and charts on Enable. He asked me to work on this project. Fortunately, I had taught myself, with a little help from my colleagues, the basics of Fortran after work and during free time. I modified the original plot output subroutine of Missile Datcom so that it contained more force and moment coefficients (aerodynamic data). I also wrote another program to organize the output by angle of attack and convert the modified plot output into an ASCII file which could be read by Enable. (fig.4) Attachment four is the modified subroutine; attachment five is my plot conversion program. output. I then developed a methodology for transfering the files from the VAX to a hard disk and from there to the Enable directory. From there, the data is accessible for graphing or other purposes. Enable is only one of several utility programs I learned to use this summer. I also learned to use Wordstar, Gem, and Exact.

As I said earlier in this paper, I learned a great deal this summer, more than I could really convey in this report. I had high expectations for this program, but they have all been surpassed, besides, who wouldn't want to play with missiles? Anyway, aeronautical engineering is now a career in which I am interested. I realize I am not impartial, but I truly believe that this program is one of our country's greatest investments for the future. I hope to return next year and continue learning and contributing.

```
2.5, 0.0,  0.0000,  0.0000,  0.4317,  0.0000,  0.0000,  0.0000,  8.2655, -1.3813, -8.2422,  1.3705,  0.0000,  0.4692,
3.0, 0.0,  0.0000,  0.0000,  0.4032,  0.0000,  0.0000,  0.0000,  7.5561, -0.9874, -7.5331,  0.9768,  0.0000,  0.4672,
3.5, 0.0,  0.0000,  0.0000,  0.3740,  0.0000,  0.0000,  0.0000,  7.0582, -0.7144, -7.0323,  0.7025,  0.0000,  0.4655,
2.5, 3.0,  0.4328, -0.0723,  0.4321,  0.0000,  0.0000,  0.0000,  8.2655, -1.3813, -8.2390,  1.3713,  0.0000,  0.4692,
3.0, 3.0,  0.3956, -0.0517,  0.4041,  0.0000,  0.0000,  0.0000,  7.5561, -0.9874, -7.5297,  0.9775,  0.0000,  0.4672,
3.5, 3.0,  0.3696, -0.0374,  0.3752,  0.0000,  0.0000,  0.0000,  7.0582, -0.7144, -7.0289,  0.7032,  0.0000,  0.4655,
```


Fig.4

ATTACHMENT ONE

Lineat Theory Input Summary

Shot Number: BS89061260 HVM-SBN 1
14-AUG-90 12:39:17

Starting Point to Fit	[LN]:	27
Length of First Fit Section	[NA]:	43
Incremental Section Increment	[ND]:	0
Summing Section Increment	[NB]:	5
Frequency Guess	[NAUTO]:	0
Roll Flag	[NROLL]:	2
Automatic Guess Flag	[NSG]:	0
Spin Flag	[NSPIN]:	1
Sections Flag	[NSECT]:	1
Shot Group Number	[IGRP]:	5

CNa	:	15.00000
Cma	:	-50.00000
Clp	:	-5.0,000
Cld	:	0.06500
CX Mach	:	-0.05000
CX?	:	0.00000
Twist (cal/rev)	:	1000.00000

Initial Estimates

			Parameter	Solve for Flag
Nutation Vector	[K1]	(deg):	1.09450	1
Precession Vector	[K2]	(deg):	2.39130	1
Nutation Damping Exp.	[L1]	(1/ft):	-0.00600	1
Precession Damping Exp.	[L2]	(1/ft):	-0.00090	1
Nutation Vector Orientation	[P1]	(deg):	184.90089	1
Precession Vector Orientation	[P2]	(deg):	107.38560	1
Nutation Frequency	[W1]	(deg/ft):	3.60320	1
Precession Frequency	[W2]	(deg/ft):	-3.73930	1
Nutation Frequency Change	[WD1]	(deg/ft**2):	0.00090	1
Precession Frequency Change	[WD2]	(deg/ft**2):	-0.00050	1
Trim Vector	[K3]	(deg):	-0.08730	1
Trim Vector Orientation	[P3]	(deg):	15.12320	1

Tunnel XYZ Input Data

Time (sec)	Down-Range Travel [X] (ft)	Horizontal Motion [Y] (ft)	Vertical Motion [Z] (ft)	Raw Roll [PHI] (deg)	Pitch [THETA] (deg)	Yaw [PSI] (deg)
0.0000000	7.1008	4.69320	2.35368	141.0	-1.301	2.345
0.0012527	11.9844	4.70402	2.33443	111.5	-0.859	1.820
0.0025581	17.0475	4.73234	2.33235	100.4	0.048	1.659
0.0051257	26.9553	4.80836	2.32975	60.8	1.295	0.543
0.0102912	46.9137	4.93050	2.29843	-22.4	2.415	-1.434
0.0128648	56.8333	4.99301	2.27071	-67.8	1.537	-1.673
0.0154672	66.8705	5.04245	2.25322	-129.2	-0.083	-1.241
0.0167662	71.8437	5.06842	2.24125	-147.8	-1.086	-0.767
0.0180575	76.8429	5.09194	2.22482	-191.5	-1.657	-0.223
0.0284607	116.8190	5.30791	2.16471	-446.4	1.080	2.058
0.0310459	126.7394	5.37642	2.15258	-528.1	2.356	0.473
0.0336725	136.7958	5.44592	2.13577	-623.0	2.334	-0.951
0.0362617	146.7137	5.50331	2.10734	-695.6	1.721	-1.724
0.0375510	151.6582	5.53148	2.09369	-745.6	0.946	-1.738
0.0454091	181.6749	5.68502	2.00967	-1007.6	-2.321	1.358
0.0467388	186.7430	5.71668	2.00461	-1039.4	-2.013	1.779
0.0559773	221.9353	5.91690	1.93703	-1414.4	2.070	0.545
0.0585999	231.8948	5.98319	1.91726	-1506.0	2.050	-0.887
0.0612199	241.8505	6.04529	1.88622	-1621.2	1.109	-1.488
0.0638661	251.8925	6.10292	1.85232	-1744.4	-0.172	-1.590
0.0664803	261.8071	6.14732	1.81771	-1849.5	-1.314	-0.621
0.0691636	271.9675	6.20097	1.79559	-1954.4	-1.792	0.468
0.0704432	276.8147	6.22724	1.78160	-2019.1	-1.813	1.132
0.0743820	291.7120	6.30150	1.75292	-2180.1	-0.599	2.385
0.0783740	306.7880	6.39724	1.72757	-2376.9	1.318	1.658
0.0823569	321.8254	6.49124	1.69463	-2546.7	1.954	0.026
0.0863133	336.7288	6.58593	1.65275	-2724.7	1.090	-1.512
0.0903911	352.0964	6.67250	1.60863	-3038.1	-1.031	-1.010
0.0983778	382.0729	6.81897	1.52705	-3293.0	-0.667	1.945
0.1023543	396.9847	6.90535	1.49527	-3360.7	1.035	1.595
0.1063655	412.0056	7.00090	1.46011	-3674.2	1.570	0.239
0.1103935	427.1081	7.08584	1.40844	-3898.9	0.830	-1.120
0.1143883	442.0076	7.16806	1.36234	-4073.8	-0.626	-0.806
0.1183699	456.8634	7.25883	1.33385	-4273.7	-1.450	0.347
0.1224189	471.9484	7.33338	1.28711	-4482.7	-0.674	1.857
0.1305139	502.0437	7.51443	1.22148	-4883.6	1.620	0.269
0.1345301	516.9609	7.60820	1.17865	-5091.0	0.998	-1.053
0.1386080	532.0734	7.68934	1.13133	-5282.5	-0.535	-0.673
0.1426500	547.0344	7.76674	1.08428	-5507.7	-1.231	0.360
0.1467094	562.0680	7.84269	1.04564	-5701.6	-0.694	1.632
0.1534628	586.9545	7.97148	0.99281	-6055.9	1.445	0.438

Linear Theory Analysis Summary : Page 1

Projectile Physical Properties

Diameter	0.449 (in)	11.40 (mm)
Weight	0.173 (lbm)	78.61 (gram)
Axial Inertia	0.005 (lbm-in2)	14.69 (gm-cm2)
Transverse [Y] Inertia	0.781 (lbm-in2)	2285.68 (gm-cm2)
Transverse [Z] Inertia	0.781 (lbm-in2)	2285.68 (gm-cm2)
Inertial Cross product	0.000 (lbm-in2)	0.00 (gm-cm2)
Length	7.954 (in)	20.204 (cm)

Center of Gravity (fraction from nose) 0.4527

Air Properties

Density	0.0023265 (slug/ft3)	0.0011990 (gm/cm3)
Speed of Sound	1129.90 (ft/sec)	344.39 (m/sec)

Reynolds Number	15296988.
Temperature (Deg C)	21.9400
Pressure (mbar)	1019.9800
Relative Humidity	0.4700

Linear Theory Analysis Summary : Page 2

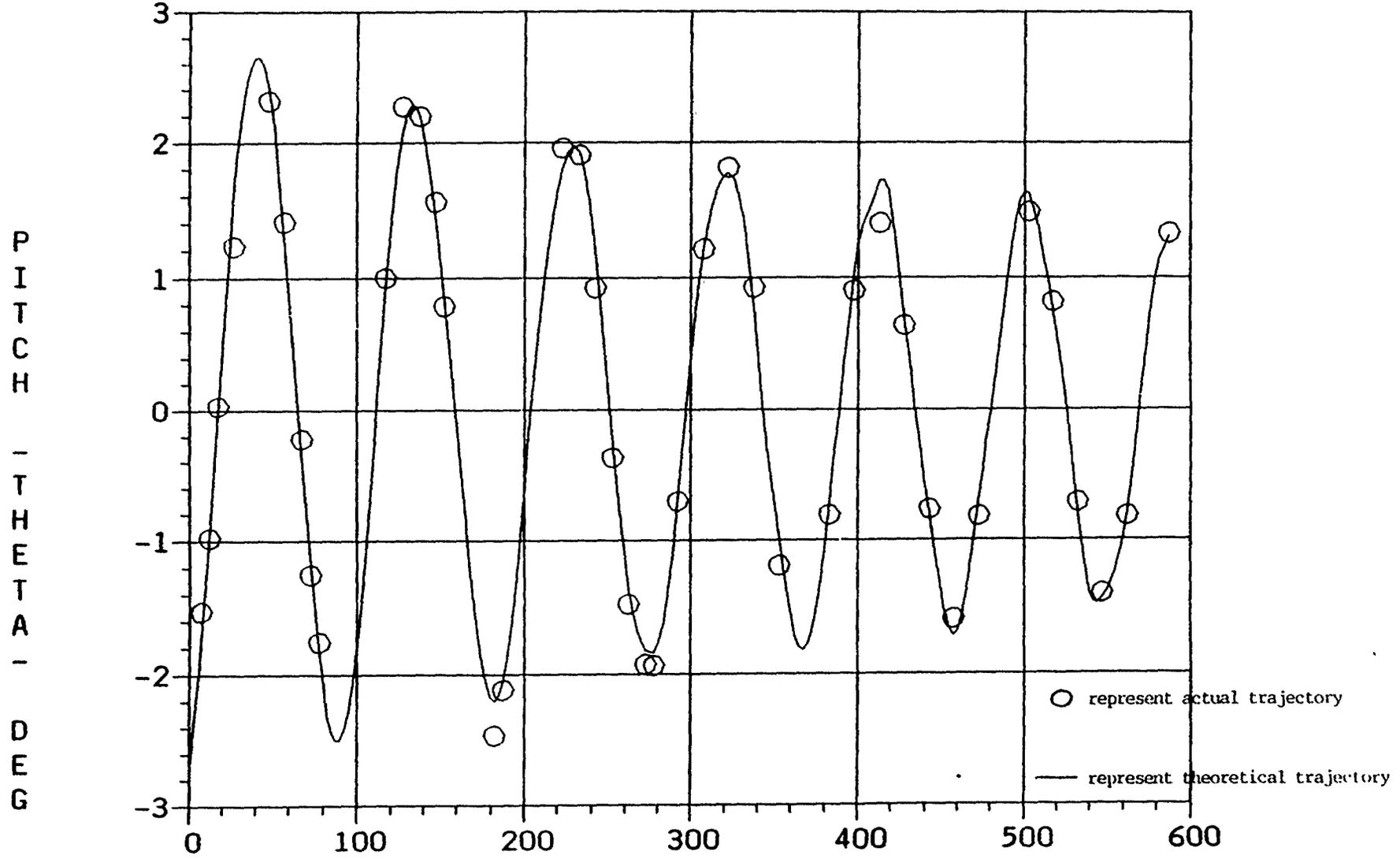
Starting point to fit	[LN]	:	1.	1.	15.
Number of Stations in Fit	[N]	:	41.	27.	37.
Length of first section to fit	[NA]	:	41.	27.	23.
Total Number of Stations	[NT]	:	41.	41.	41.
Nutation Vector	[K1]	(deg):	1.0925	1.0554	1.3584
Precession Vector	[K2]	(deg):	2.3875	2.2939	2.7270
Nutation Damping Exponent	[L1]	(1/ft):	-0.006009	-0.006063	-0.007453
Precession Damping Exponent	[L2]	(1/ft):	-0.000865	-0.000487	-0.001237
Nutation Vector Orientation	[PHI1]	(deg):	184.67	182.20	337.65
Precession Vector Orientation	[PHI2]	(deg):	107.43	108.03	78.65
Nutation Frequency	[W1]	(deg/ft):	3.6125455	3.6195209	2.7237592
Precession Frequency	[W2]	(deg/ft):	-3.7388017	-3.7610002	-3.5816219
Nutation Freq. Change	[WD1]	(deg/ft2):	0.0009038	0.0009825	0.0031944
Precession Freq. Change	[WD2]	(deg/ft2):	-0.0004871	-0.0003443	-0.0008795
Trim Vector	[K3]	(deg):	0.0902890	0.1236945	0.0304601
Trim Vector Orientation	[PHI3]	(deg):	146.2218018	140.4029999	155.4788818
Swerve Constant	[S1]	:	0.0104036	0.0067725	-0.0071807
Swerve Constant	[S2]	(1/ft):	0.0141149	0.0136648	0.0149663
Roll Constant	[R1]	:	147.8267	132.9821	547.8830
Roll Constant	[R2]	(1/ft):	-3.5722	-2.9904	-7.3852
Roll Constant	[R3]	(1/ft**2):	-0.0199798	-0.0237385	-0.0087608
Probable Error in Distance	[PE-X]	(ft):	0.0074827	0.0064576	0.0060750
Probable Error in Angle	[PE-A]	(deg):	0.1070419	0.1025765	0.0772555
Probable Error in Swerve	[PE-S]	(ft):	0.0046812	0.0031093	0.0029504
Probable Error in Time	[PE-T]	(msec):	1.9801	1.6909	1.6147
Probable Error in Roll	[PE-R]	(deg):	21.2827	5.1457	27.2999
Distance to First Fit Station	[X-LN]	(ft):	7.1008	7.1008	181.6749
Distance to Last Fit Station	[X-N]	(ft):	586.9545	336.7288	516.9609
Del Bar Squared	[DBSQ]	:	3.7688	4.7741	3.3079
	[DBEF]	:	5.2325	6.9586	4.8131
	[DBEP]	:	3.5242	4.7832	3.2110

Linear Theory Analysis Summary : Page 3

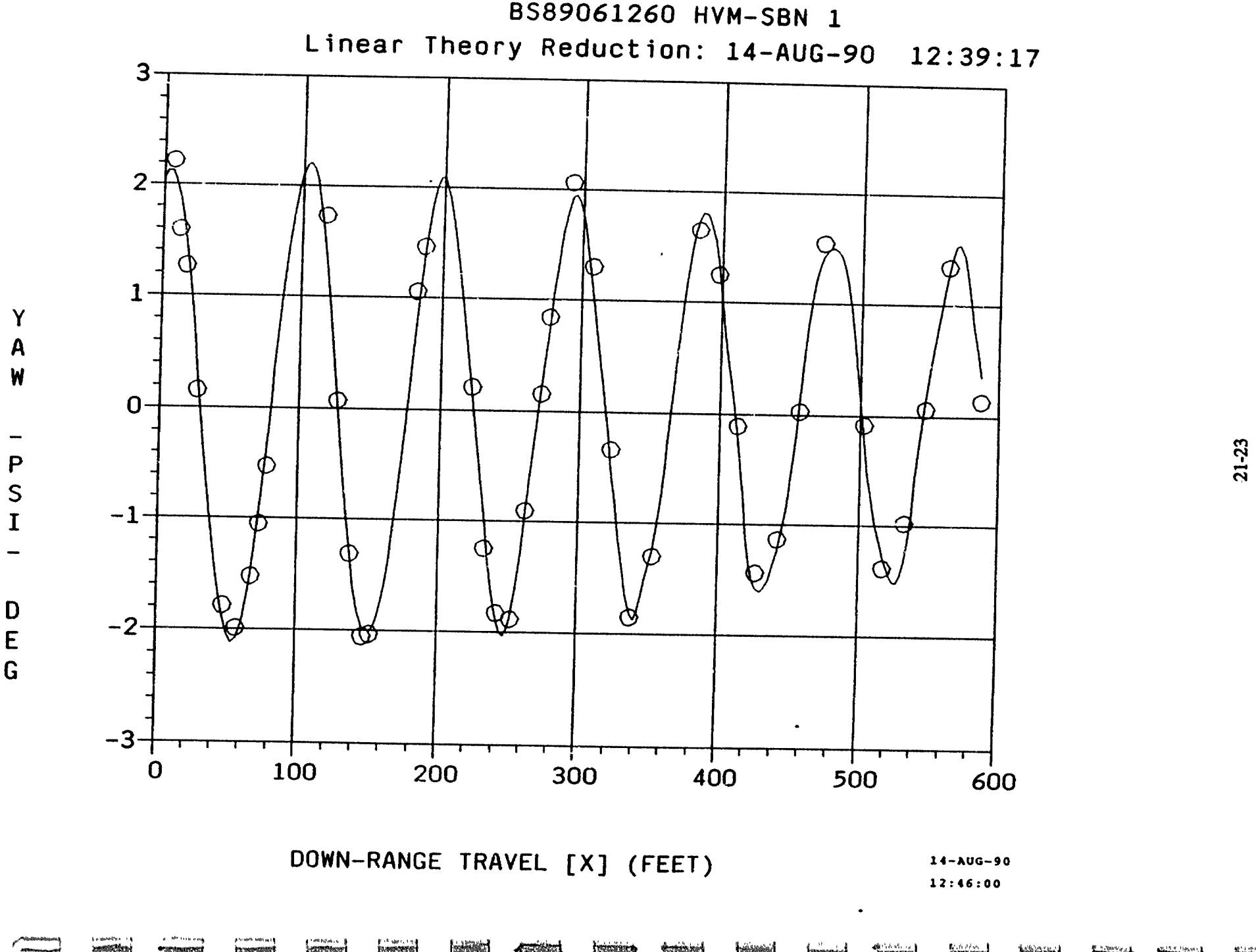
Mach Number	[MACH]	:	3.3445	3.3800	3.3298
Cma	[CMA]	:	-16.1529	-15.5116	-15.9232
Cmq	[CMQ]	:	-625.4003	-564.2821	-967.9614
Cnpa	[CNPA]	:	4454.0093	706.8575	442.5691
CD	[CD]	:	0.3554	0.3559	0.3538
CD0	[CD0]	:	0.3475	0.3461	0.3473
CD Mach	[CDM]	:	0.3536	0.3559	0.3513
CD0 Mach	[CDM0]	:	0.3457	0.3461	0.3448
CNa	[CNA]	:	6.9030	6.7432	6.4410
Roll Computed Clp	[CLPR]	:	1.6134	3.5882	1.0083
Frequency Computed Clp	[CLPW]	:	-93.1840	-21.4038	-42.8217
Gyro Stability	[GYRO]	:	0.0000	0.0000	-0.0001
Dynamic Weight Factor	[TAU]	:	-0.0007	-0.0048	-0.0085
Velocity at Mid-Range	[VMID]	(ft/sec):	3778.9390	3819.0596	3762.2996
Mid-Range Distance	[XMID]	(ft):	297.0277	171.9148	349.3179
Reference Distance	[XREF]	(ft):	7.1008	7.1008	7.1008
Reference Mach Number	[MREF]	:	3.3800	3.3800	3.3800
Yaw of Repose	[YREP]	(deg):	0.0000027	0.0000182	0.0000331
Initial Velocity	[VZRO]	(ft/sec):	3872.0664	3871.7793	3871.3140

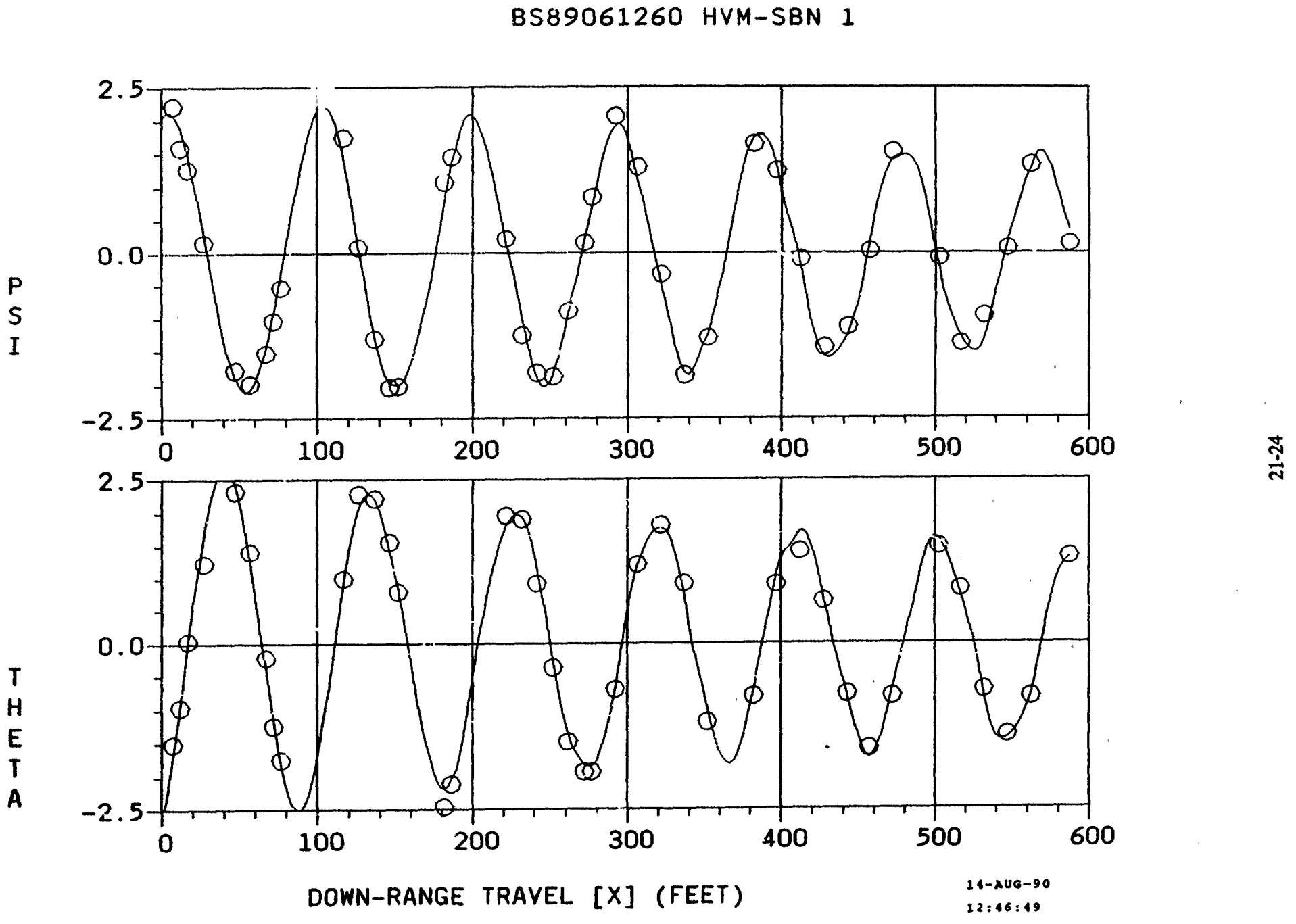
BS89061260 HVM-SBN 1

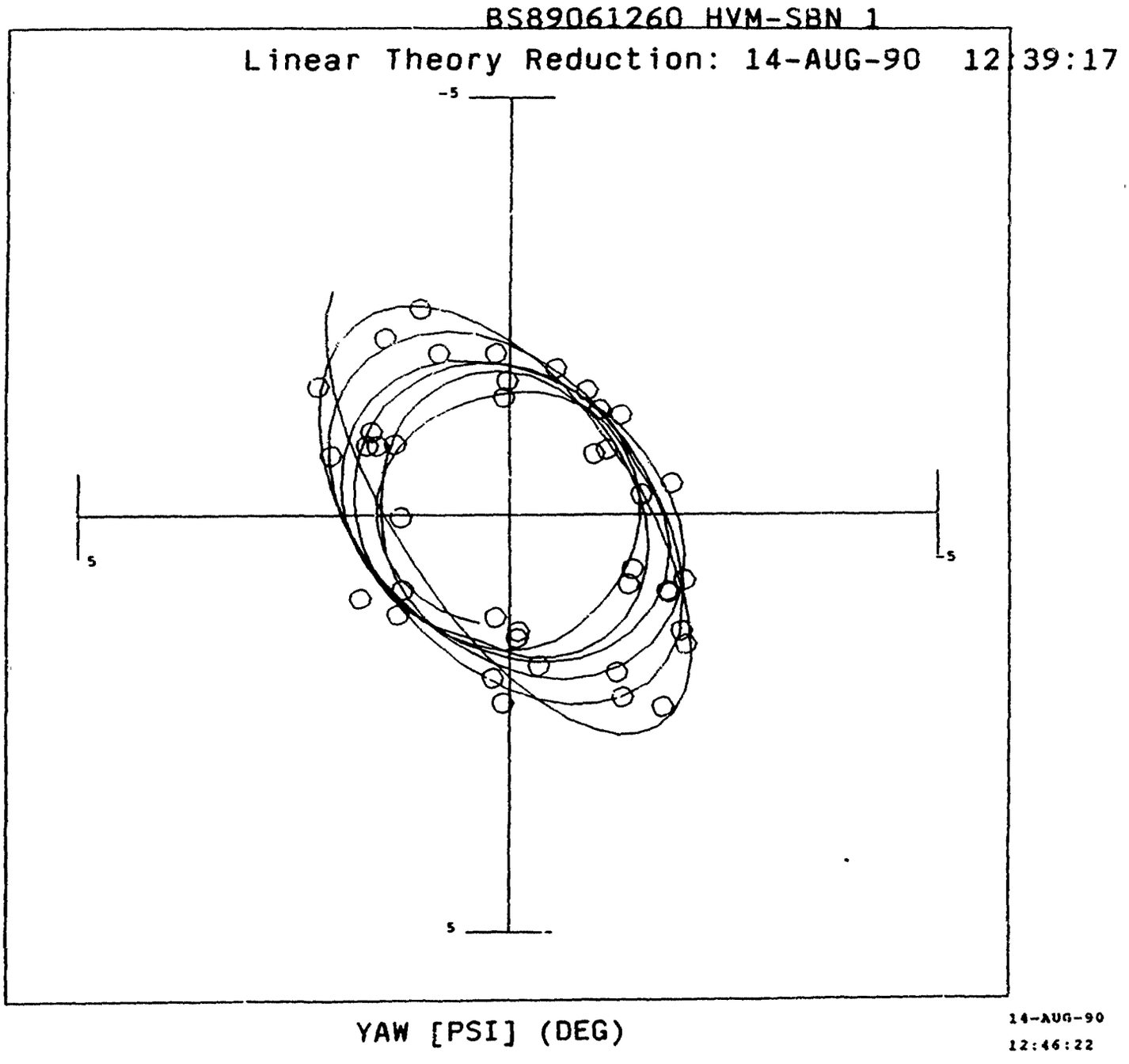
Linear Theory Reduction: 14-AUG-90 12:39:17



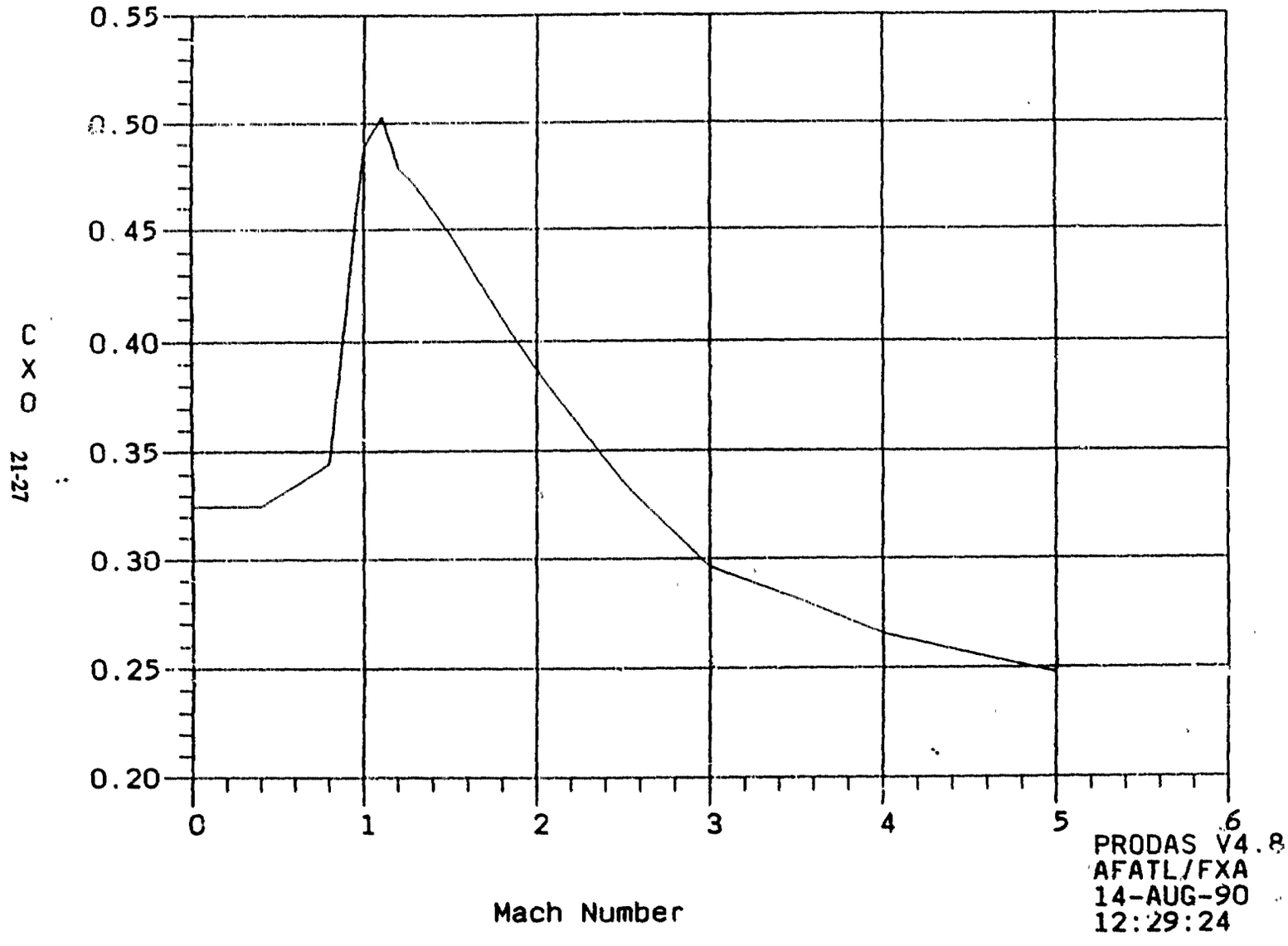
14-AUG-90
12:45:45


BS89061260 HVM-SBN 1
Linear Theory Reduction: 14-AUG-90 12:39:17
YAW - PSI - DEG
3
2
1
0
-1
-2
-3
0
100
200
300
400
500
600
DOWN-RANGE TRAVEL [X] (FEET)
14-AUG-90
12:46:00


BS89061260 HVM-SBN 1
PSI
THETA
2.5
0.0
-2.5
0
100
200
300
400
500
600
DOWN-RANGE TRAVEL [X] (FEET)
14-AUG-90
12:46:49

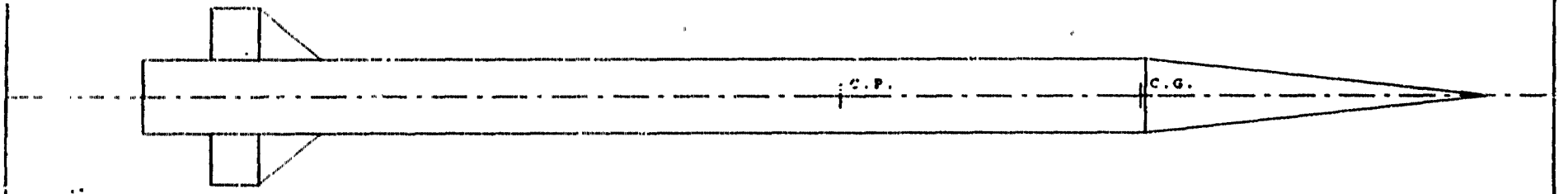

BS89061260 HVM-SBN 1
Linear Theory Reduction: 14-AUG-90 12:39:17
-5
5
5
PITCH - THETA - DEG
YAW [PSI] (DEG)
14-AUG-90
12:46:22

ATTACHMENT TWO


0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
C
X
0
0
1
2
3
4
5
6
Mach Number
PRODAS V4.8
AFATL/FXA
14-AUG-90
12:29:24

PRODAS V4.10 Filename: HVM 14-AUG-90

HYPERVELOCITY MSSILE
Projectile Length = 7.960 (in)
Ogive Length = 2.025 (in)



Stability Results

Rifling Twist	0.00	(in/rev)	0.0 (calibers/rev)
Muzzle Velocity	5000.00	(ft/sec)	
Aircraft Velocity	0.00	(ft/sec)	
Air Density	.002376	(slugs/ft**3)	
Air Temperature	59.0	(Deg F)	
Air Temperature	59.0	(Deg F)	
Center of Pressure from Nose	3.823	(in)	4.496 (calibers)
Center of Gravity from Nose	2.057	(in)	4.571 (calibers)
Dynamic Stability Factor	0.000		
Static Margin	1.766	(in)	3.925 (calibers)

Filename: HVM 14-AUG-90 :HYPERVELOCITY MISSILE

Enter projectile physical properties

1 - Number of Fins	:	4
2 - Fin Type (1 Rectangular 2 Delta)	:	3
(3 Clipped Delta 4 Swept)		
3 - Nose Shape (1 Conical 2 Spike)	:	1
4 - Projectile Diameter (inch):		0.450
5 - Axial Inertia (lbm-in**2):		0.00302
6 - Transverse Inertia (lbm-in**2):		0.59490
7 - Weight (lbm):		0.129
8 - Projectile Velocity (ft/sec):		5000.
9 - Magnus Moment Coefficient :		0.000
10 - Roll Induced Side Moment Coefficient :		0.000
11 - Temperature (Deg F):		59.000
12 - Gun Bore Diameter (inches):		0.000
13 - Any T Pads on Fins (0 - no 1 - yes):		0.
14 - Screen print out (0 - no 1 - yes):		1.

Filename: HVM 14-AUG-90 :HYPERVELOCITY MSSILE

		(inches)	(calibers)
1 - Boattail Length	:	0.000	0.000 - 21
2 - Boom Length	:	0.000	0.000 - 22
3 - Boom or Aft Boattail Diameter	:	0.450	1.000 - 23
4 - Nose to CG Distance	:	2.057	4.571 - 24
5 - Projectile Cylindrical Length	:	5.935	13.189 - 25
6 - Projectile Base to Fin Trailing Edge Distance	:	0.400	0.889 - 26
7 - Fin Root Chord	:	0.657	1.460 - 27
8 - Fin Tip Chord	:	0.290	0.644 - 28
9 - (or fin angle) (deg)	:	40.187	40.187 - 29
10 - Fin Height	:	0.310	0.689 - 30
11 - Fin Thickness	:	0.015	0.033 - 31
12 - Fin Lead Thickness	:	0.014	0.032 - 32
13 - Nose Radius	:	450.	1000. - 33
14 - Meplat Diameter	:	0.054	0.120 - 34
15 - Nose Length	:	2.025	4.500 - 35

Filename NVM 14-AUG-90

	Cyl. Body Length	Ogive Length	Boattail Length	Boom Length	C.G. from Nose	Band Diameter	Meplat Diameter	Boom or Aft Boattail Dia.	Ogive Radius
in inches	7.960	2.025	0.000	0.000	2.057	0.000	0.054	450.000	
in calibers	13.189	4.500	0.000	0.000	4.571	0.000	0.120	1000.000	

Diameter (in)	Weight (lbm)	Gun Bore (in)	Temperature (Deg F)	Air Density (slugs/ft**3)	Axial Mom. (lbm-in**2)	Trans. Mom. (lbm-in**2)
0.450	0.129	0.000	59.000	0.002376	0.003020	0.594900

	Fin Root Chord	Fin Tip Chord	Fin Height	Fin Thickness	Fin Lead Thickness	Base to Trailing Edge Distance
in inches	0.657	0.290	0.310	0.015	0.014	0.400
in calibers	1.460	0.644	0.689	0.033	0.032	0.889

Projectile has 4 clipped delta fins

Aerodynamic Coefficients

Mach	CD	Cna	CPN	Cma	Cmq	CXB	CXF	CnaB	CnaF	CPNB	CPNF
0.010	0.325	8.322	11.875	-60.784	-2312.	0.282	0.042	2.993	5.329	4.569	15.978
0.400	0.325	8.805	12.059	-65.930	-2422.	0.282	0.042	2.993	5.812	4.562	15.919
0.800	0.344	9.878	12.427	-77.549	-2683.	0.285	0.060	3.013	6.859	4.539	15.892
1.000	0.489	10.493	12.506	-83.267	-2808.	0.410	0.079	3.059	7.435	4.346	15.864
1.100	0.502	10.559	12.887	-87.803	-2919.	0.426	0.077	2.928	7.631	4.640	16.051
1.200	0.479	10.623	12.860	-88.046	-2940.	0.407	0.072	2.980	7.642	4.572	16.091
1.300	0.470	10.406	12.730	-84.905	-2889.	0.396	0.072	3.033	7.373	4.495	16.110
1.500	0.447	10.193	12.472	-80.533	-2835.	0.375	0.072	3.158	7.034	4.284	16.149
1.800	0.409	9.747	12.084	-73.228	-2716.	0.339	0.070	3.285	6.462	4.045	16.170
2.000	0.387	9.324	11.752	-66.956	-2599.	0.316	0.071	3.371	5.953	3.939	16.177
2.500	0.335	8.244	10.854	-51.799	-2302.	0.264	0.071	3.486	4.759	3.580	16.183
3.000	0.296	7.324	10.068	-40.256	-2065.	0.225	0.071	3.450	3.873	3.197	16.188
3.500	0.281	6.570	9.457	-32.097	-1863.	0.209	0.072	3.400	3.170	3.182	16.188
4.000	0.265	5.985	8.899	-25.902	-1707.	0.192	0.073	3.350	2.635	3.167	16.188
5.000	0.247	5.216	8.055	-18.169	-1491.	0.174	0.074	3.250	1.965	3.137	16.188

Epsilon = 25.25989
Lambda = 0.42056

Filename: HVM 14-AUG-90 HYPERVELOCITY MISSILE

Stability Parameters of Statically Stable Missiles

P (rad/sec)	GYRO	SIGMA	RATE	P/W1	PD/2V	W1 (rad/sec)	W2 (rad/sec)	L1 (1/ft)	L2 (1/ft)	AF	YR (deg)
46.247	0.000	3930.079	-2.001	0.100	0.000	461.410	-461.175	-0.0067	-0.0067	1.005	0.000
231.234	0.000	786.016	-2.003	0.501	0.001	461.879	-460.706	-0.0067	-0.0067	1.318	0.000
369.974	0.000	491.261	-2.004	0.800	0.001	462.232	-460.354	-0.0067	-0.0067	2.609	0.000
416.221	0.000	436.677	-2.005	0.900	0.002	462.350	-460.237	-0.0067	-0.0067	4.258	0.000
439.344	0.000	413.694	-2.005	0.950	0.002	462.409	-460.178	-0.0067	-0.0067	5.818	0.000
462.467	0.000	393.009	-2.005	1.000	0.002	462.467	-460.120	-0.0067	-0.0067	6.890	0.000
485.591	0.000	374.295	-2.005	1.050	0.002	462.526	-460.061	-0.0067	-0.0067	5.514	0.000
508.714	0.000	357.281	-2.006	1.100	0.002	462.585	-460.003	-0.0067	-0.0067	3.852	0.000
554.961	0.000	327.508	-2.006	1.199	0.002	462.703	-459.886	-0.0067	-0.0067	2.135	0.000
693.701	0.000	262.007	-2.008	1.498	0.003	463.056	-459.535	-0.0067	-0.0067	0.791	0.000
924.935	0.000	196.507	-2.010	1.995	0.003	463.646	-458.951	-0.0067	-0.0067	0.332	0.000
1387.402	0.000	131.006	-2.015	2.985	0.005	464.827	-457.784	-0.0067	-0.0066	0.125	0.000
2312.337	0.000	78.608	-2.026	4.949	0.009	467.198	-455.461	-0.0067	-0.0066	0.042	0.000
4624.674	-0.001	39.314	-2.052	9.774	0.017	473.179	-449.704	-0.0068	-0.0066	0.010	0.000
9249.349	-0.003	19.676	-2.107	19.057	0.035	485.364	-438.414	-0.0069	-0.0064	0.003	0.000
13874.022	-0.006	13.138	-2.165	27.868	0.052	497.847	-427.422	-0.0070	-0.0063	0.001	0.000
27748.045	-0.023	6.626	-2.355	51.666	0.104	537.062	-396.212	-0.0074	-0.0059	0.000	0.000

Stability Analysed for

Velocity (ft/sec)	CD	CNa	Cma	Cmq
9000.	0.257	5.618	-22.212	-1603.861

ATTACHMENT THREE

FOLLOWING ARE THE CARDS INPUT FOR THIS CASE

```
SOSE
DIM IN
DERIV RAD
 $FLTCON NMACH=3.,MACH=2.5,3.,3.5,
         NALPHA=2.,ALPHA=0.,3.,
         ALT=0.,$
 $REFQ XCG=3.66,LREF=7.96,SREF=.15904,RHR=255.,$
 $AXIBOD TNOSE=1., LNOSE=2.025, LCENTR=5.935, DNOSE=.450,
         DCENTR=.45,$
 $FINSET1 CHORD=.657,.290,
          SSPAN=.225,.535,
          SWEEP=0., STA=1.,
          XLE=6.903,
          NPANEL=4.,
          LMAXU=2*.00333,LFLATU=2*.99334,ZUPPER=2*.0075,LER=2*.5,$
PLOT
PRINT AERO BODY
NEXT CASE
```

THE BOUNDARY LAYER IS ASSUMED TO BE TURBULENT OVER ALL COMPONENTS OF THE CONFIGURATION

THE INPUT UNITS ARE IN INCHES, THE SCALE FACTOR IS 1.0000

FLIGHT CONDITIONS								REFERENCE DIMENSIONS				
MACH NUMBER	ALTITUDE FT	VELOCITY FT/SEC	PRESSURE LB/IN**2	TEMPERATURE DEG R	REYNOLDS NUMBER 1/FT	SIDESLIP ANGLE DEG	ROLL ANGLE DEG	REF. AREA IN**2	REF. LENGTH LONG. IN	LAT. IN	MOMENT LONG. IN	REF. CENTER VERTICAL IN
3.00	0.00	3348.80	1.470E+01	5186.70	2.120E+07	0.00	0.00	0.159	7.960	7.960	3.660	0.000

ALPHA	CA-FRICTION	CA-PRESSURE/WAVE	CA-BASE	CA-ALPHA
0.000	0.16846	0.04877	0.09500	0.00000
3.000	0.16869	0.04943	0.09500	0.00000

NOTE - THE BASE DRAG INCREMENT IS NOT INCLUDED IN THE AXIAL FORCE CALCULATIONS

CROSS FLOW DRAG PROPORTIONALITY FACTOR = 1.00000

ALPHA	CN-POTENTIAL	CN-VISCOUS	CM-POTENTIAL	CM-VISCOUS	CDC
0.000	0.0000	0.0000	0.0000	0.0000	0.2800
3.000	0.1668	0.0247	0.0447	-0.0020	0.4370

FLIGHT CONDITIONS

MACH NUMBER	ALTITUDE FT	VELOCITY FT/SEC	PRESSURE LB/IN**2	TEMPERATURE DEG R	REYNOLDS NUMBER 1/FT	SIDESLIP ANGLE DEG	ROLL ANGLE DEG
3.50	0.00	3906.94	1.470E+01	5186.70	2.474E+07	0.00	0.00

REFERENCE DIMENSIONS

REF. AREA IN**2	REF. LENGTH LONG. IN	REF. LENGTH LAT. IN	MOMENT REF. CENTER LONG. IN	MOMENT REF. CENTER VERTICAL IN
0.159	7.960	7.960	3.660	0.000

ALPHA	LONGITUDINAL CN	LONGITUDINAL CM	LONGITUDINAL CA	LATERAL DIRECTIONAL CY	LATERAL DIRECTIONAL CLN	LATERAL DIRECTIONAL CLL
0.00	0.000	0.000	0.374	0.000	0.000	0.000
3.00	0.370	-0.037	0.375	0.000	0.000	0.000

DERIVATIVES (PER RADIAN)

LONGITUDINAL CNA	LONGITUDINAL CMA	LATERAL DIRECTIONAL CYB	LATERAL DIRECTIONAL CLNB	LATERAL DIRECTIONAL CLLB
7.058E+00	-7.144E-01	-7.032E+00	7.025E-01	-6.670E-09
7.058E+00	-7.144E-01	-7.029E+00	7.032E-01	-2.254E-05

ALPHA	CL	CD	CL/CD	X-C.P.
0.00	0.000	0.374	0.000	-0.101
3.00	0.349	0.394	0.887	-0.101

PANEL DEFLECTION ANGLES (DEGREES)

FIN SET	FIN 1	FIN 2	FIN 3	FIN 4
1	0.00	0.00	0.00	0.00

LINEAR DATA FOR BODY ALONE WAS GENERATED USING THE SECOND-ORDER SHOCK EXPANSION METHOD

ATTACHMENT FOUR

```
      SUBROUTINE PLOT3(IC,IM,CN,CM,CA,CY,CLN,CLL,
     !                 CNA,CMA,CYB,CNB,CLB,XCP)
C
C***  SUBROUTINE TO GENERATE A PLOT FILE ON TAPE UNIT 3
C***  FOR POST PROCESSING BY A PLOTTING PROGRAM
C
C***  INPUTS
C
C     IM -- CURRENT VALUE OF MACH NUMBER (INTEGER VALUE)
C     CN -- ARRAY OF NORMAL FORCE DATA
C     CM -- ARRAY OF PITCHING MOMENT DATA
C     CA -- ARRAY OF AXIAL FORCE DATA
C     CY -- ARRAY OF SIDE FORCE DATA
C    CLN -- ARRAY OF YAW MOMENT DATA
C    CLL -- ARRAY OF ROLL MOMENT DATA
C    CNA -- ARRAY OF NORMAL FORCE DERIVATIVE WITH ALPHA
C    CMA -- ARRAY OF PITCHING MOMENT DERIVATIVE WITH ALPHA
C    CYB -- ARRAY OF SIDE FORCE DERIVATIVE WITH BETA
C    CNB -- ARRAY OF YAW MOMENT DERIVATIVE WITH BETA
C    CLB -- ARRAY OF ROLL MOMENT DERIVATIVE WITH BETA
C
      COMMON /CASEID/ IDCASE(74),KOUNT,NAMSV(100),CASE,NOEXTR,NOLAT,IR
      COMMON /REFQN/  SREF,LREF,LATREF,ROUGH,XCG,ZCG,SCALE,BLAYER
      COMMON /FLC/    NALPHA,ALPHA(20),BETA,PHI,NMACH,MACH(20),
     1                ALT,REN(20),VINF(20),TINF(20),PINF(20)
      COMMON /CONST/  PI,RAD,UNUSED,KAND
      COMMON /LOGIC/ LDMPCS,LDAMP,LBUILD,LNACA,LDERDG,
     1               LDERRD,LPART,LNAME,LPLOT,
     2               LFLT,LREFQ,LAXIS,LFIN1,LFIN2,LFIN3,LFIN4,
     3               LDEFL,LTRIM,LDIMIN,LDIMFT,LDIMCM,LDIMM
      LOGICAL        LDMPCS,LDAMP,LBUILD,LNACA,LDERDG,
     1               LDERRD,LPART,LNAME,LPLOT,
     2               LFLT,LREFQ,LAXIS,LFIN1,LFIN2,LFIN3,LFIN4,
     3               LDEFL,LTRIM,LDIMIN,LDIMFT,LDIMCM,LDIMM
      DIMENSION CN(20),CM(20),CA(20),CY(20),CLN(20),CLL(20)
      DIMENSION CNA(20),CMA(20),CYB(20),CNB(20),CLB(20)
      DIMENSION XCP(20)
C
      REAL NALPHA,NMACH,MACH,LREF,LATREF
C
      INTSGER PIN,PFT,PCM,PM,CASE
C
      DATA PIN / 4HI   /
      DATA PFT / 4HF   /
      DATA PCM / 4HC   /
      DATA PM  / 4HM   /
C
      IR=IR+1
      NALP=NALPHA+0.5
C
      IPTYP=PFT
      IF(LDIMIN)IPTYP=PIN
      IF(LDIMCM)IPTYP=PCM
      IF(LDIMM )IPTYP=PM
C
      CALL CVRTFT
C
      YCG = 0.0
C
      IF(IM.EQ.1)WRITE(11,1030)LREF,XCG
C
      DO 1000 I=1,NALP
         WRITE(3,1040)MACH(IM),ALPHA(I),CN(I),CM(I),CA(I),CY(I),
     1   CLN(I),CLL(I),CNA(I),CMA(I),CYB(I),CNB(I),CLB(I),XCP(I)
 1000 CONTINUE
C
```

```
C     WRITE(3,1050)
C
      CALL CVRTUS
C
 1010 FORMAT(3HRUN,I4,I3,2X,A1,25A1)
 1020 FORMAT(F10.4,F10.0,F10.4)
 1030 FORMAT(2(F10.4))
 1040 FORMAT(2F4.1,12F10.4)
 1050 FORMAT(1HR)
      RETURN
      END
```

ATTACHMENT FIVE

```
      PROGRAM PLTFORM
C   A PROGRAM TO GENERATE ENABLE READY PLOT OUTPUT
      DIMENSION AA(25,25)
1000  FORMAT(A26)
1010  FORMAT(I3)
1020  FORMAT(2F10.4)
1040  FORMAT(2(F4.1,1H,),12(F9.4,1H,))
1050  FORMAT(2F4.1,12F10.4)
1060  FORMAT(A45)
C
C
      REAL LREF
      INTEGER NR,NC
      DIAM=0.
      NR=0
      NC=0
      LREF=0.
      XCG=0.
C
C   A SERIES OF QUESTIONS ESTABLISHING THE PARAMETERS OF THE FILE
C
      WRITE(*,1000),'INPUT NUMBER OF COLUMNS'
           READ(6,1010)NC
      WRITE(*,1000),'INPUT NUMBER OF ROWS'
           READ(6,1010)NR
      WRITE(*,1060),'INPUT DIAMETER OF MISSILE (CASE DIMENSIONS)'
           READ(6,1020)DIAM
C
      READ(11,1020)LREF,XCG
C
C   READING  INTO AN ARRAY
      DO 20 I=1,9999
      READ(3,1050,END=25)(AA(I,J),J=1,NC)
      AA(I,14)=(((AA(I,14)*DIAM)-XCG)/LREF*-1.0)
 20   CONTINUE
C
C
C   FINDING THE NUMBER OF ALPHAS
C
 25   IF (NR.GT.I-1) NR=I-1

      DO 50 K=2,NR
           IF(AA(1,2).EQ.AA(K,2))GO TO 60
 50   CONTINUE
C
C
C   ORGANIZING AND WRITING ACCORDING TO ALPHAS
C
 60   L=K-1
      X=FLOAT(NR/L)
      INT=NINT(X)
      DO 70 N=1,L
           DO 80 NN=0,INT-1
                WRITE(7,1040)(AA((NN*L)+N,J),J=1,NC)
 80        CONTINUE
 70   CONTINUE
C
 40      END
```

Bibliography

Anderson, John D, Jr. Introduction to Flight. 2nd ed. New York: McGraw-Hill Book Co., 1985.

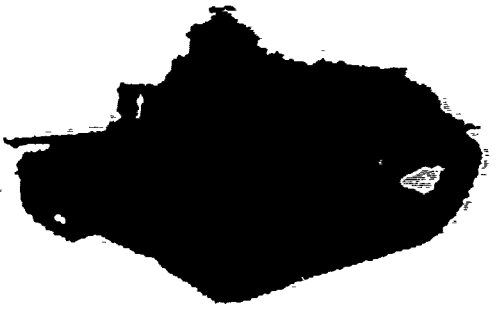
Borse, G.J. Fortran 77 and Numerical Methods for Engineers. Boston: PWS Engineering, 1985.

Eglin Computer Network User's Guide. January 25, 1988.

Optical Processing:

Digital Imagery Aquisition and Analysis

by



Danielle Walker

Niceville High School



AGA/IR

Mentor: Dennis Goldstein

Acknowledgements

I would like to thank

Dr. Dennis Goldstein

for the two summers that he spent as my mentor where he gave me invaluable guidance , knowledge and confidence.

Optics Lab

Mike VanTassle
Mac McCormick
Voncile Houston
Linda Lau
David Onuffer
Tom Speed

AGA

Lynn Deibler
Tom Davis
Otto Martinez
Jeff Barnes
Phyllis Williams
David Gray
Maggie Fenton
Fred Gibson
Randy Gove
Lisa Collins

Image Processing Lab

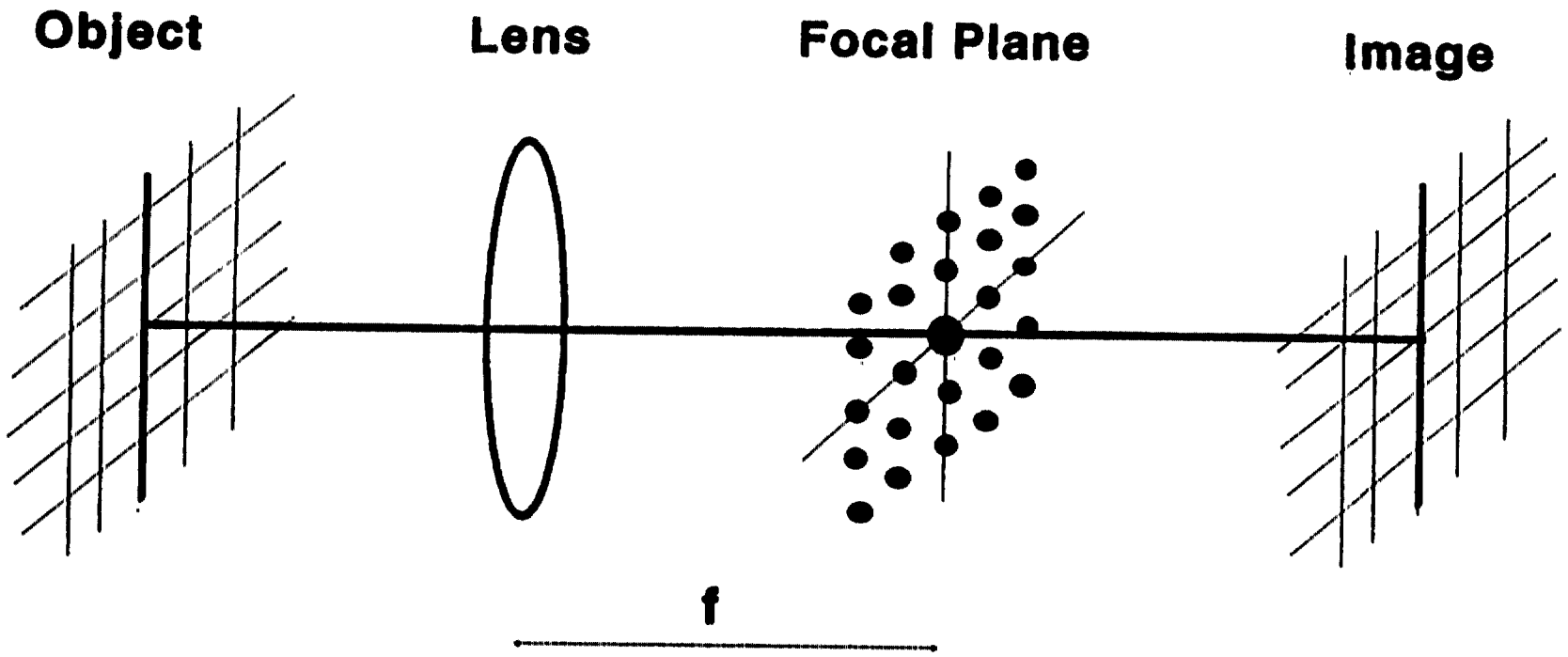
Larry Neal
Ben-z Lawrence
Steve Halprin
David Kerr
Noreen Porter

INTRODUCTION

With new advancements in technology come problems that are difficult to solve. These problems must be put into a different format that is simpler to work with. Fourier analysis is an example of a technique to simplify these problems, and the heart of Optical Processing.

Fourier synthesis techniques can be said to have begun with the first intentional manipulation of the spectrum of an image. The experiments done by Abbe, in 1893, and Porter, in 1906, demonstrate the basic principles of Fourier analysis, shown in Figure 1.

An object of fine wire mesh was illuminated by a coherent light source. The Fourier spectrum of the periodic mesh appears in the back focal plane of the imaging lens. In the image plane, the various Fourier components passed by the lens are recombined to form a replica of the mesh. When various obstructions (such as an iris, slit, or small stop) are placed in the focal plane, it becomes possible to directly manipulate the spectrum of the image in a variety of ways.



The Abbe-Porter experiment

FIGURE 1

BACKGROUND

The chart in Figure 2 illustrates the different projects that I am involved in. Dr. Dennis Goldstein is the head of the team to explore and use Optical Pattern recognition. A contractor, the University of North Texas, is developing aspect invariant spatial filters to be used in these experiments. These projects must be fed by two other system projects, Digital Imagery Production and Image Analysis, which were the projects I worked on. The Image Analysis work also feeds the Optical Design project. The Optical Design and the Spatial Filter work go into creating the actual hardware for Optical Pattern Recognition, the Optical Correlator.

Optical Processing can be demonstrated by the design in Figure 3. First, a laser gives a coherent light source. This light is sent through the image to be optically processed. As this beam passes through the image, the image is read onto the beam itself. As the beam passes through the first lens a Fourier transform is performed. The spatial frequency distribution of the image, which looks merely like a pattern of dots, appears at the focal plane of the lens. A second high quality lens will reconstruct the image so that the resultant image is shown in the second focal plane. This is known as the Four F Processor because the length of the series is four times the focal length of the lens.

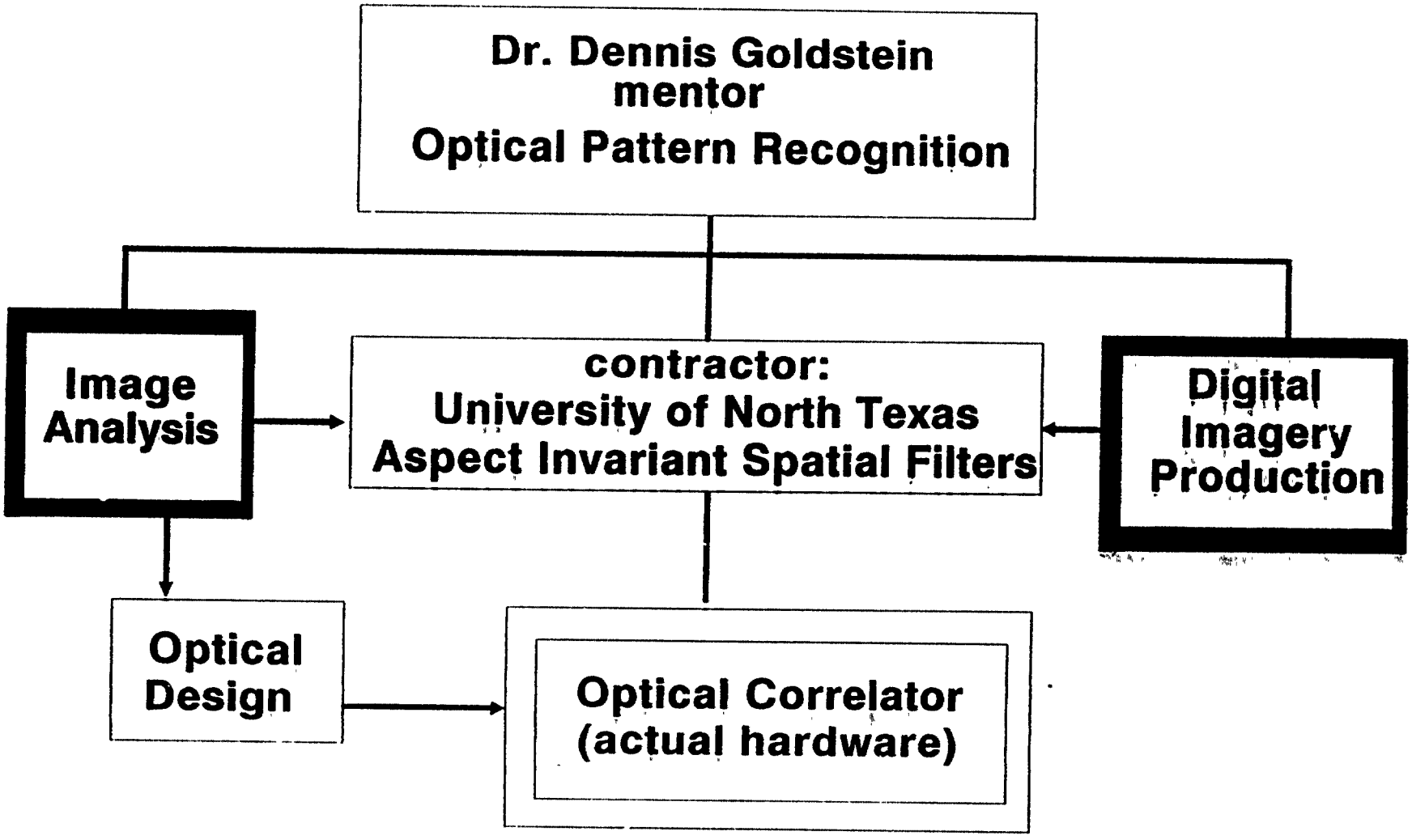

Dr. Dennis Goldstein
mentor
Optical Pattern Recognition
Image Analysis
contractor:
University of North Texas
Aspect Invariant Spatial Filters
Digital Imagery Production
Optical Design
Optical Correlator
(actual hardware)

FIGURE 2

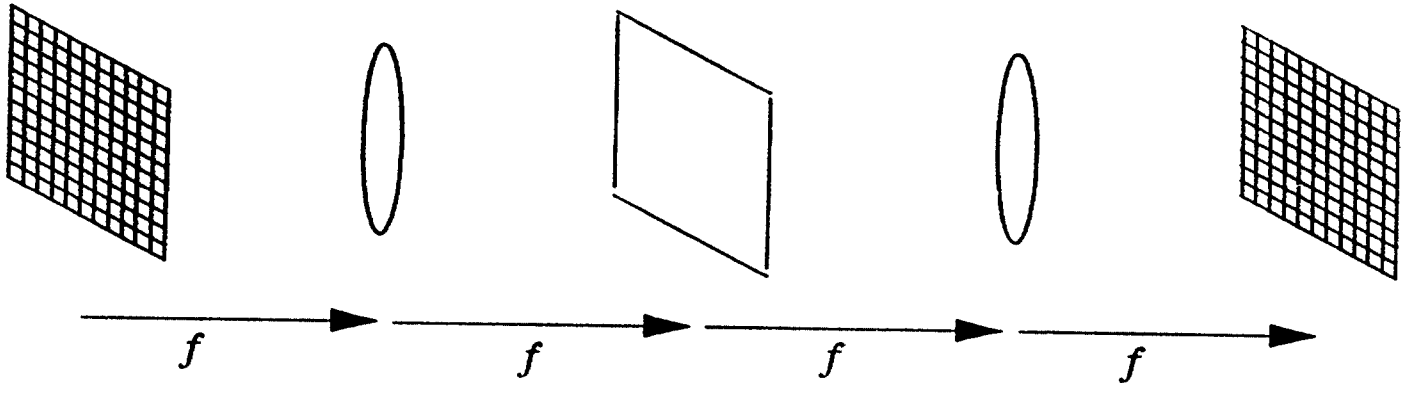


image plane *Focal plane* *Resultant plane*

FIGURE 3

Description of Research

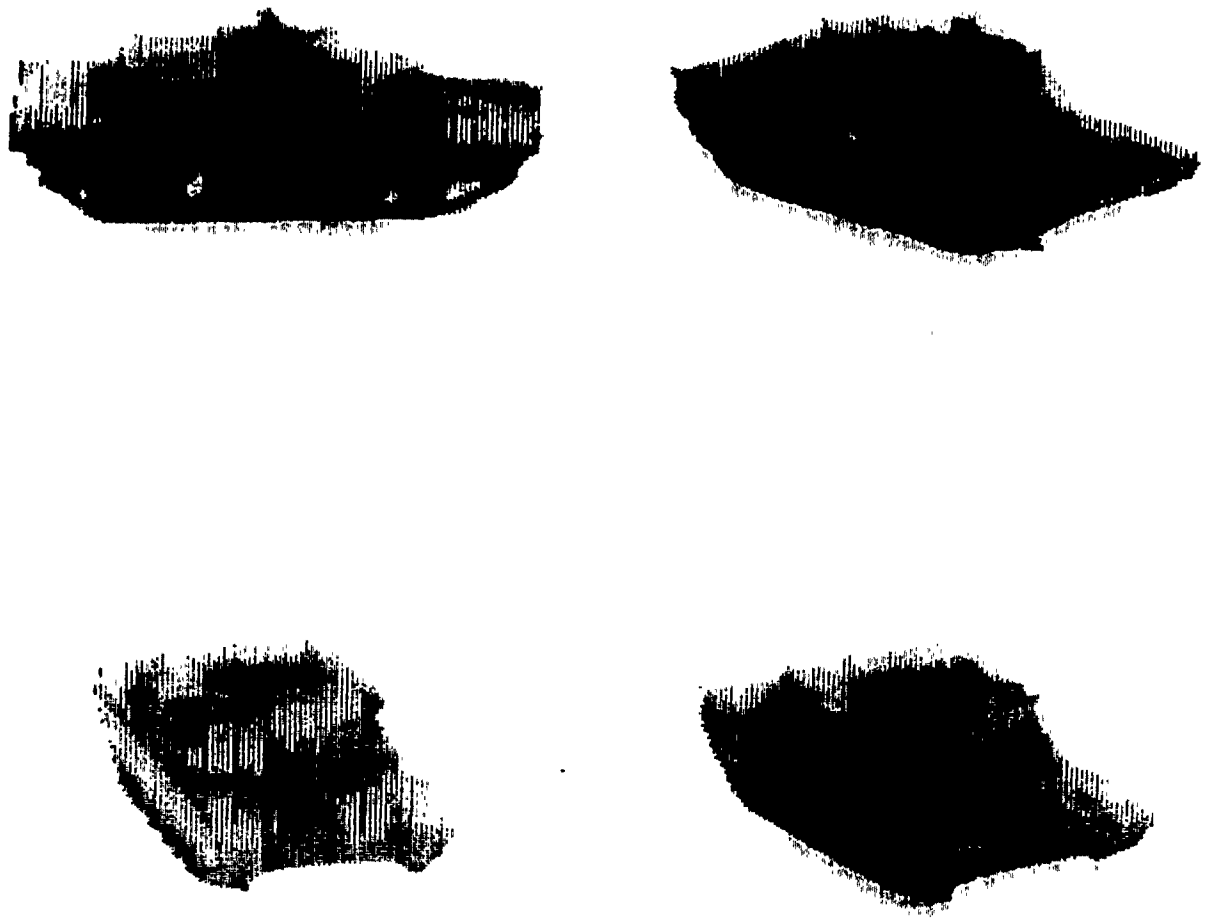
The Digital Imagery work was all computer simulation of actual optical occurrences. Instead of a live image, data arrays are used. To gather the necessary data arrays, I needed to accumulate different target images onto computer files. For these images I used the following models:

Tanks:	M1
	M48
	M60
	T-62
	M113 APC
Airplanes:	F-19
	F-15
	F-14
	Herrier
SDI vehicles:	Post Boost Vehicle
	Bus

Some of these target images are shown in Figures 4-10. Figure 11 shows that these models were set up in the Optics Laboratory on a stabilized table with the camera at a twenty degree depression angle to simulate a normal vector for a air-to-ground missile.

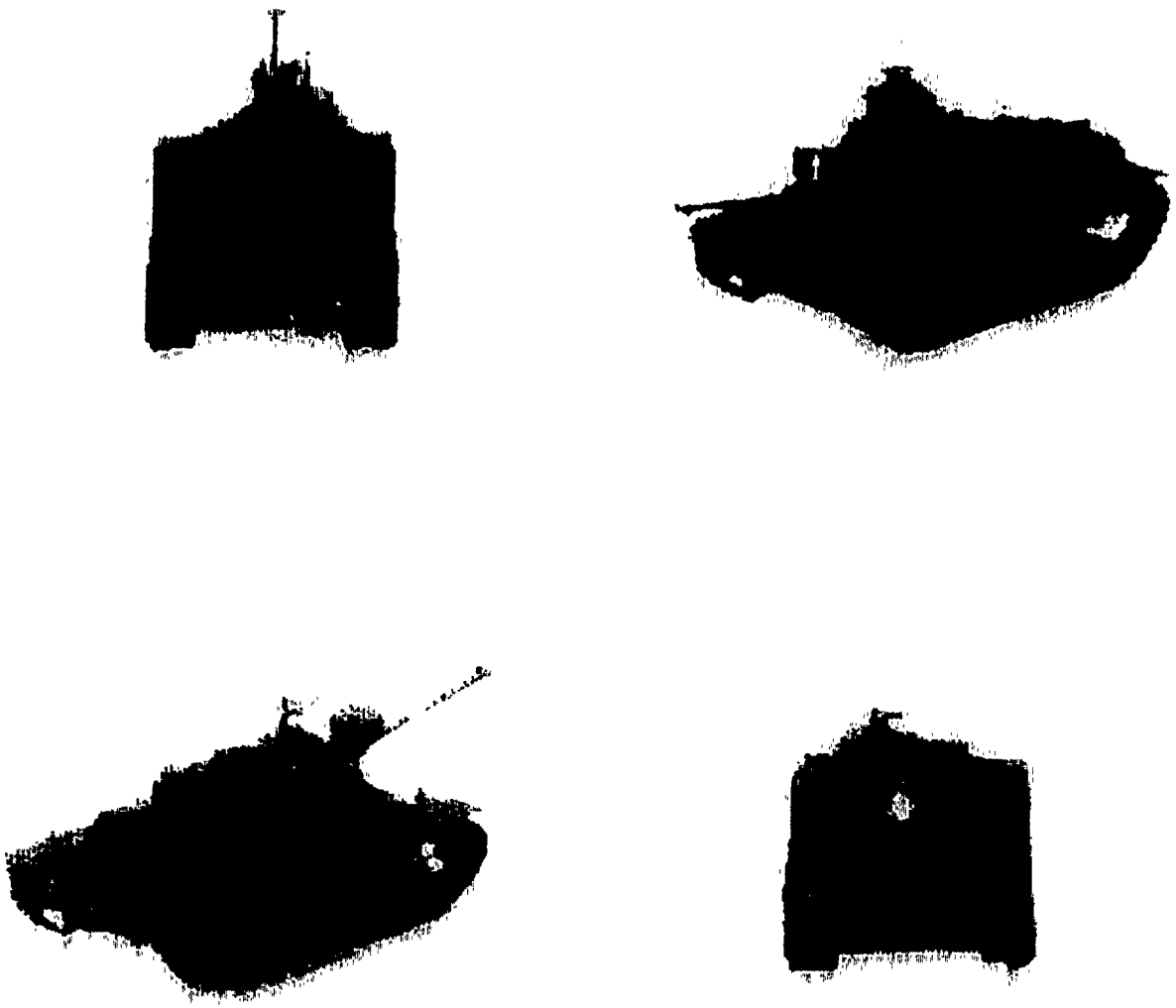
The control box for the camera had it's horizontal and vertical syncs directly linked to the VMS VAX 8650 system. The video hookup was between the camera itself and the VAX. In the Image Processing Laboratory, I captured the live image sent through the VAX onto one of the DeAnza monitors. The monitors were controlled through the input/output terminals, the VT 220 or the VT 340. The set up is shown in Figure 12.

While the image was on the screen, the Cellular Array Processing System (CAPS) was used to save and manipulate the image. CAPS is a high-level, image language which is enables the observer to view the imagery on the monitor as a user's algorithm



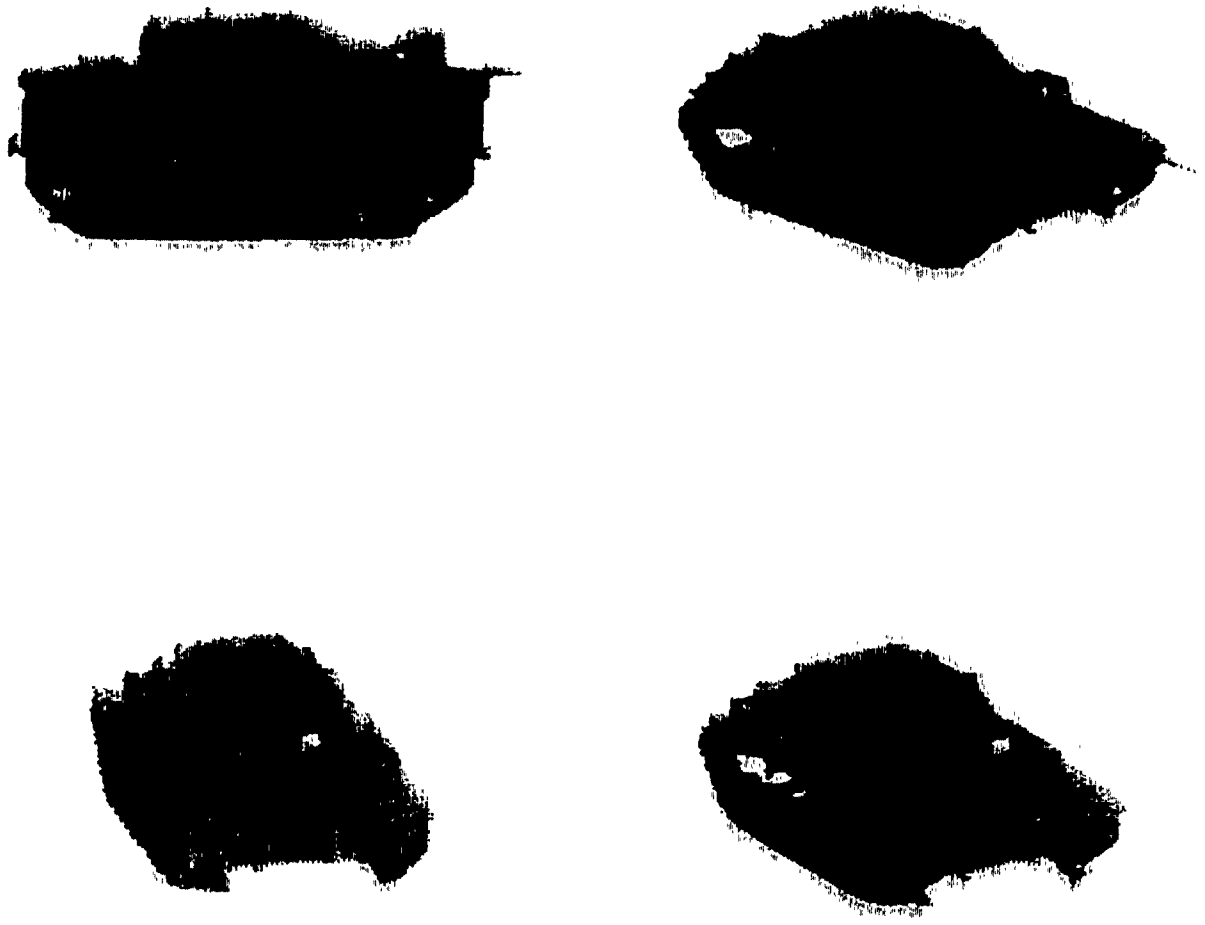
M1 TANK

FIGURE 4



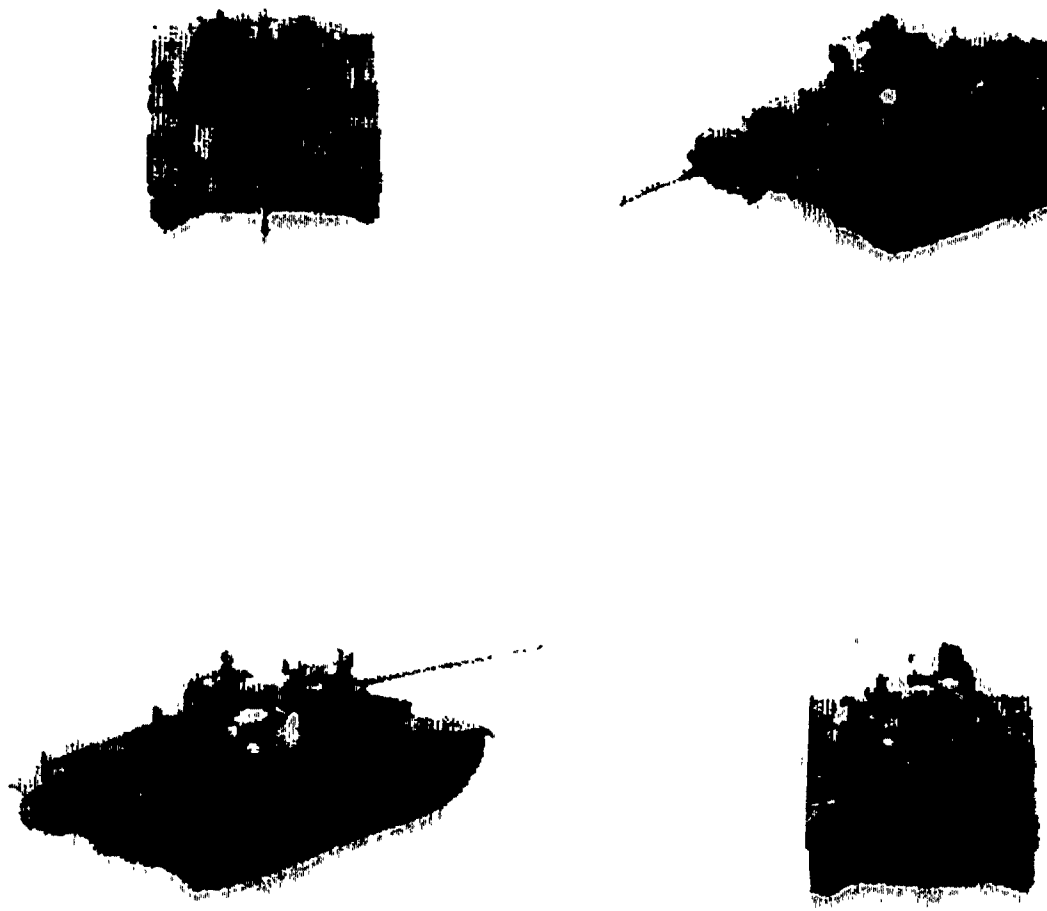
48

FIGURE 5



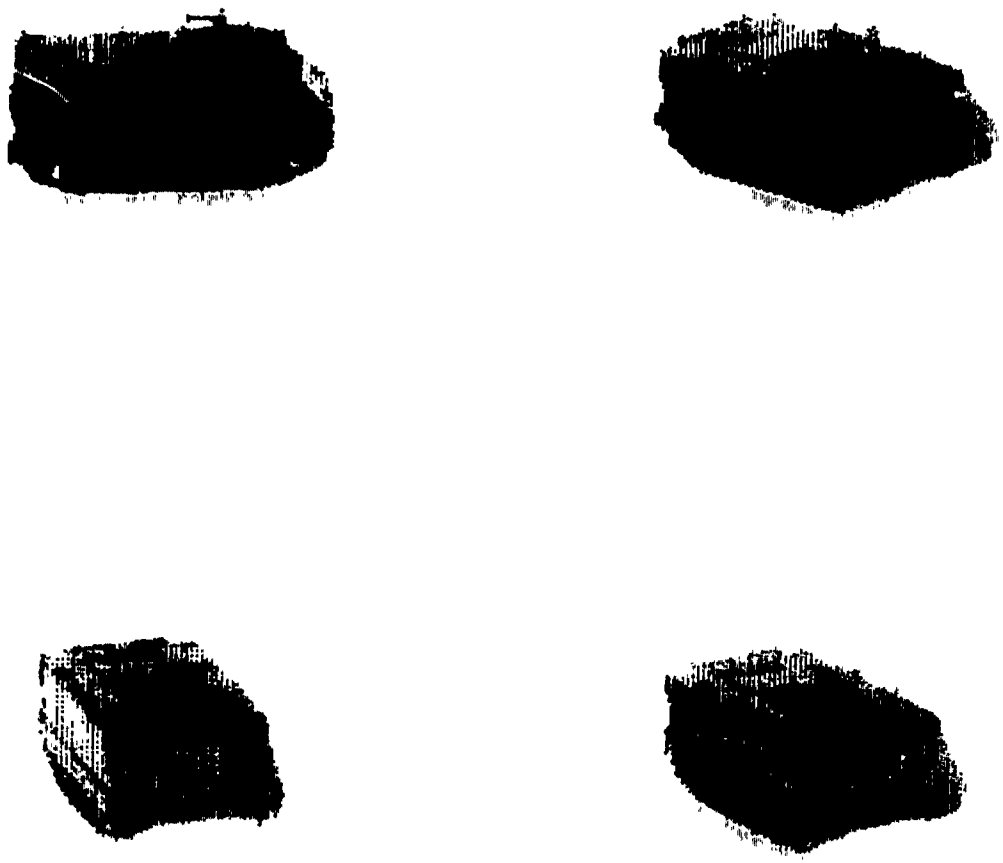
1160

FIGURE 6



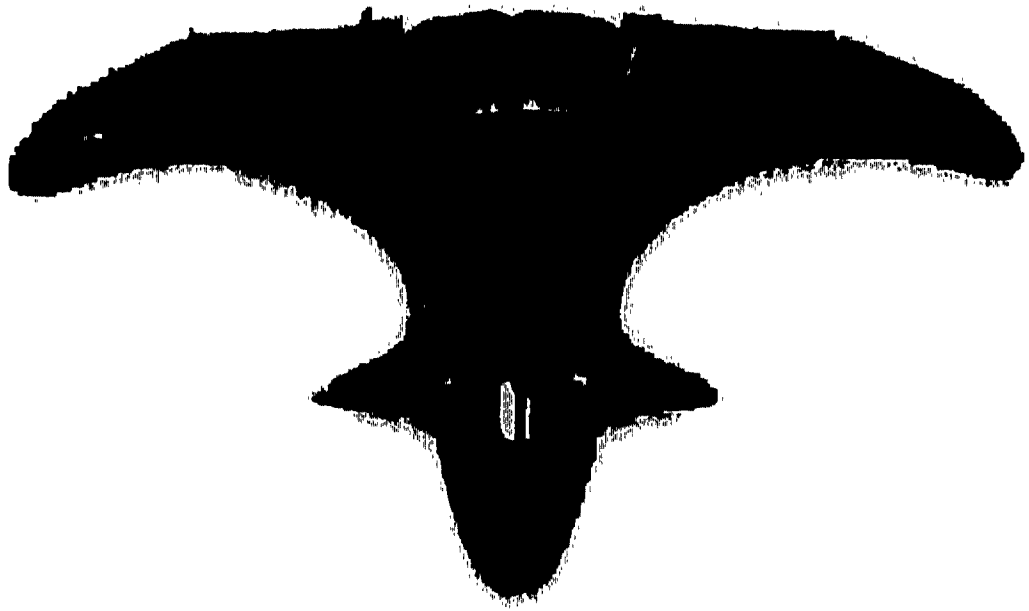
T-62

FIGURE 7



M113 APC

FIGURE 8



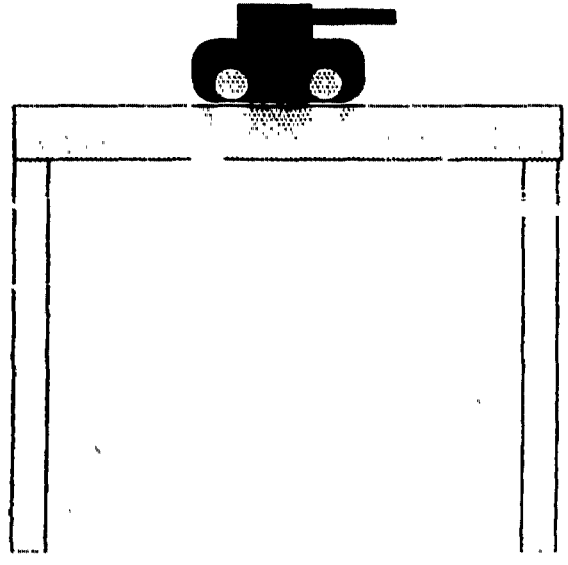
F19

FIGURE 9



BUS

FIGURE 10



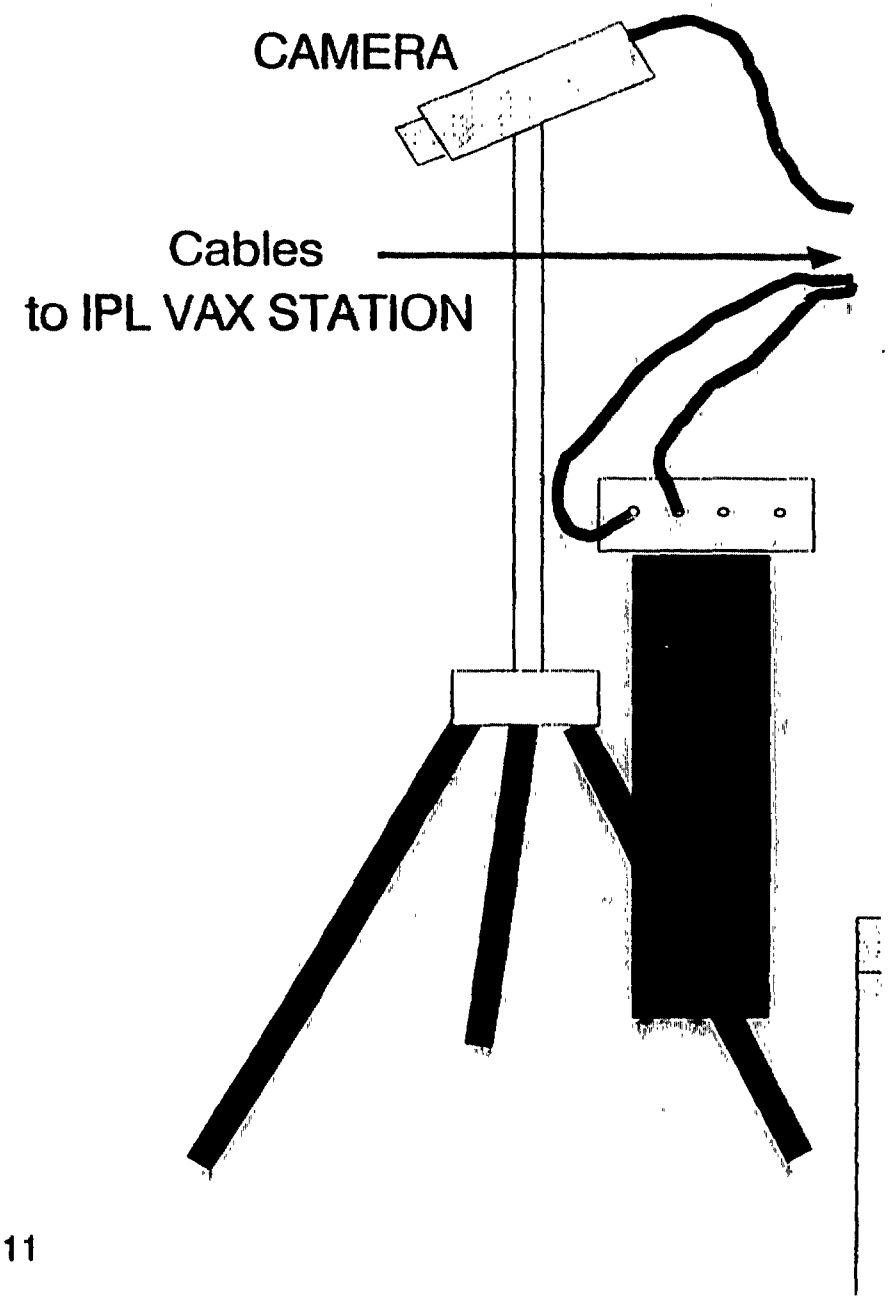


FIGURE 11

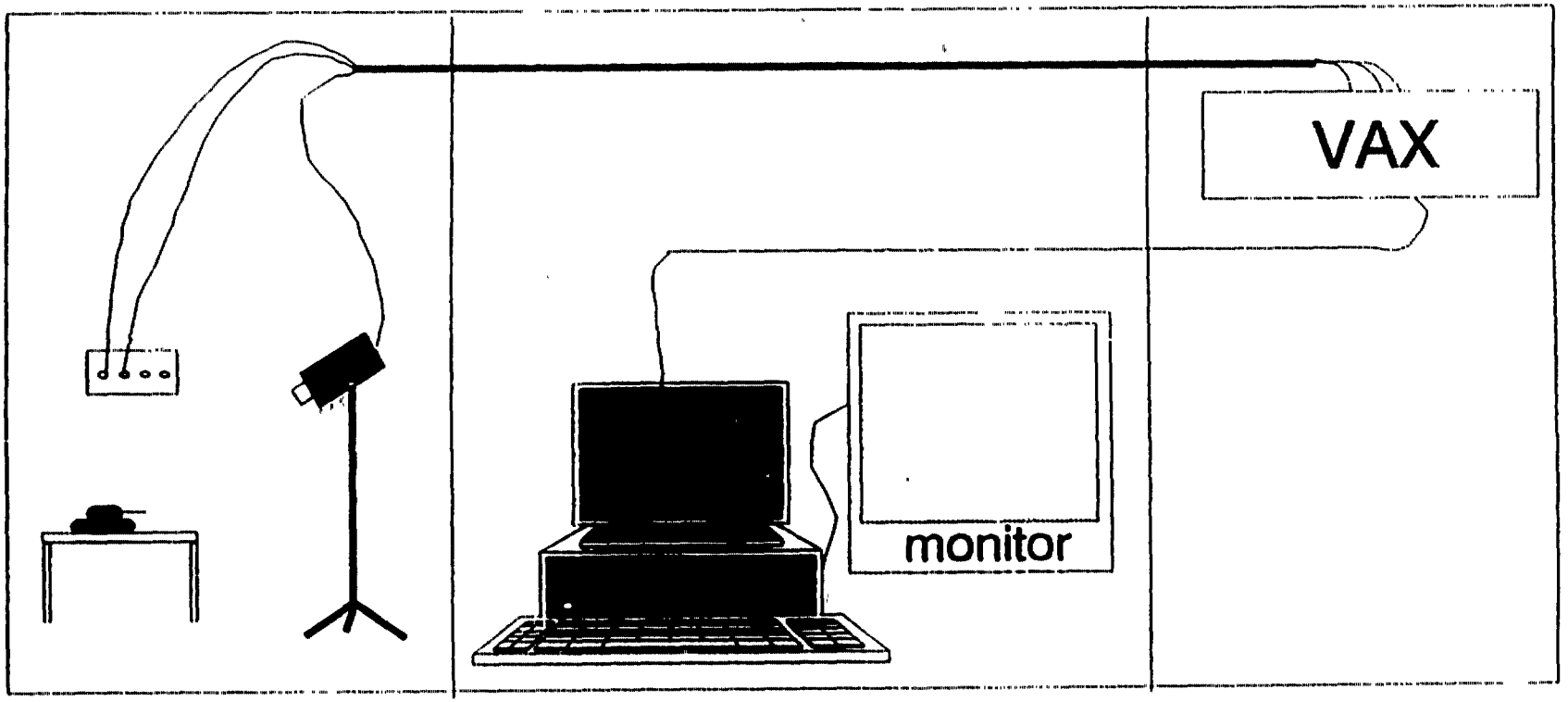


FIGURE 12

is executed. This feature comes in useful when there is limited storage space because the original image can be kept and resultant images are made from it.

The models are rotated and digitized through different sets of aspect angles. The tanks are rotated a total of two hundred twenty-five degrees, the planes are rotated a total of one hundred eighty, and the SDI vehicles are rotated a total of ninety degrees. After the model is rotated throughout all of its angles, the image files are transferred to a nine-track magnetic tape.

Many things, such as the camera arrangement, stability effects, and lighting conditions, made beginning digital imagery difficult. Because of the trial and error, I wrote a manual that outlines various techniques to make digital imagery more efficient for someone continuing this research. The CAPS portion of this manual is shown on the following page.

How to use CAPS in the IPL.

1. Log in

2. Assign the monitor to use
   A. To see which monitors are open type SHOW DEV EPA
      1. If it is on-line, then it can be used.
      2. If it is allocated, then another user already has it.
   B. After you know which monitor to use, i.e. EPA1 then type
      1. ASSIGN EPA1 IP0
      2. If you do not want anyone interfering, allocate the monitor by: ALLOCATE EPA1
      3. To make certain that the monitor is assigned:
         A. SYSINT which initilizes the screen with cursors.
         B. CUROFF which clears the screen.

3. Get into CAPS by typing CAPS
   A. To digitize an image, type:
      1. INIT which initilizes the screen in CAPS
      2. A/D which takes the picture

4. To save in CAPS
   A. To only save an object in the middle of the screen, the command would be: PICK 01 128 128 256 256
      1. TEMP01.TMP would be the name of the object. It only allows for two decimal place holders; from 00 to 99.
      2. The first 128 is where the x(row) coordinate would begin to save. The second 128 is where the y(column) coordinate would begin to save.
      3. The 256 256 is the dimension of the box to be saved.
   B. To save the entire picture, the command would be: PICK 02 0 0 512 512
      1. The name chosen for this example was TEMP02.TMP.
      2. The 0 0 would be the upper left corner and the 512 512 would be the lower right corner.

5. To unsave a picture on the monitor
   A. To reproduce a picture described in 4A, the command would be: UNSAVE -1 1 256 256 TEMP01.TMP
   B. To reproduce a picture described in 4B, the command would be: UNSAVE -1 1 512 512 TEMP02.TMP

6. Frequently used features
   A. CLR clears the screen on the monitor.
   B. COVER changes the colors of the background to the object and vice-versa; i.e. COVER 255 0
   C. COLOR 1 is the red image that is digitized
      COLOR 2 is the image transposed into black and white
      COLOR 2 1 shows a color bar at the bottom of the black and white picture.
   D. ZOOM is the size of the image
      ZOOM 2 is the image enlarged by half
      ZOOM 4 is the image enlarged by a quarter
      ZOOM 8 is the image enlarged by an eighth
   E. Instead of typing in EXIT to exit CAPS, just press Control Y.

Most of the image manipulation was done in CAPS. The first change was to switch the color scheme to black and white from the red to green hues, shown in Figure 13, found in the main CAPS coloring process.

The maximum image size is 512 x 512 pixels. Instead of saving the whole screen, a rectangular subset of the screen which contained the model was selected. This saved memory by giving the file a smaller image to hold while the whole model was in the center rectangle the excess noise was cut out.

After all of the image manipulation is done, the model is rotated one degree.

My image analysis work was mostly through the computer to simulate optimum conditions. This can be more difficult that the real optics work, but it is more reliable and efficient. In real optics work, the only time an image can manipulated is in the focal plane. Certain amplitudes and frequencies represented by dots can be filtered out to make the picture clearer or merely to single out part of the resident image. This is done by a spatial light modulator or a filter.

The three major ways to make filters are with photographic film, electron beam etcher on chrome coated glass, or a modulator. A modulator is a three dimensional, programmable mask which can be changed to fit each image. These are expensive though and are only efficient if used long term or if they are used to hold a library of images.

To generate a filter the computer needs the digital imagery in an array form. The imagery coming from the camera was in an

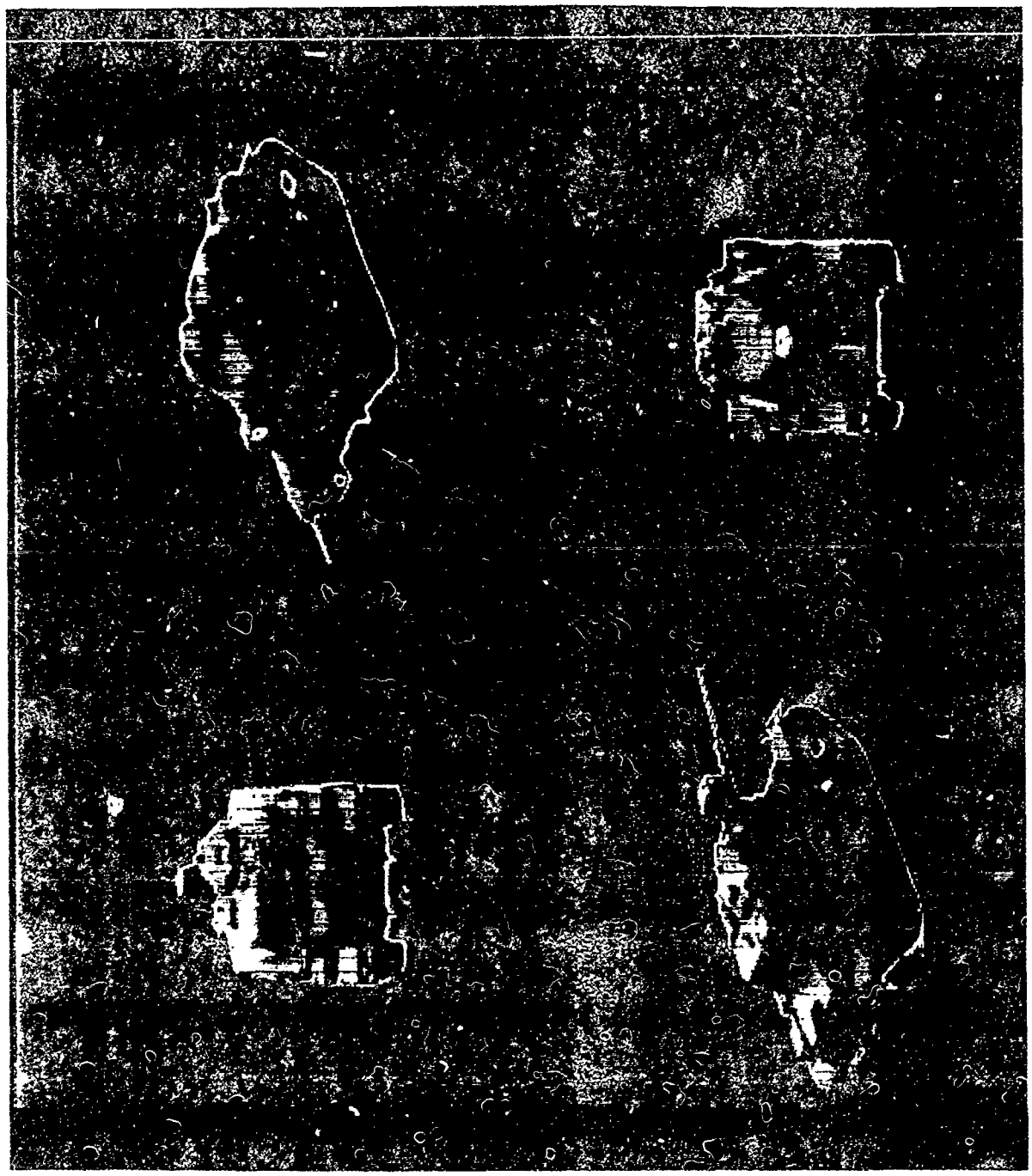


image array form. So I wrote a program that converts the image array into a data array. I wrote this program in Fortran, which is shown in Appendix 1. This program is combined with another, already-developed program which takes the two-dimensional Fourier transform on the supplied data array. This, in turn, gives a complex number represented by sixteen bits which represents the Fourier Spectrum.

I tested my program with the image of a simple black square because the Fourier transform of a black square is known and easily recognizable. I then proved that my program was effective.

Conclusions

The two projects, Digital Imagery Production and Image Analysis, support and contribute to Optical Pattern Recognition research and development. The prime setup for the Optical Pattern Recognition configuration is shown in Figure 14. The incoherent light of a scene is brought through a seeker dome, in this case transparent. The output light is changed to coherent light by a single converter, which takes the place of two separate converters used in the past. This was developed by Otto Martinez, at AFATL. With the aid of a laser, the image is read to the beam which passes through a lens. This lens takes the Fourier transform of the image. The Fourier transform, which is a representation of spatial frequency distribution and amplitude, is compared to transforms of other scenes in the target library in the programmable filter. Another lens transforms the image back into a recognizable image pattern and is shown in the detector array.

The prime goal of this research is a guidance system to be used in seekers. Optical Correlators are very useful in seekers because it gives the missile, information on target location based on the complete spatial target signature.

Some developed optical correlators have already been tested. Self-contained Optical Correlators have a laser diode and detector kept inside a glass arrangement. This Optical Correlator fits inside a seeker with the electrical controls and detector systems. Other types of seekers have the laser, detector, filters, and lenses separated by air in the actual setup of the seeker.

OPTICAL PATTERN RECOGNITION

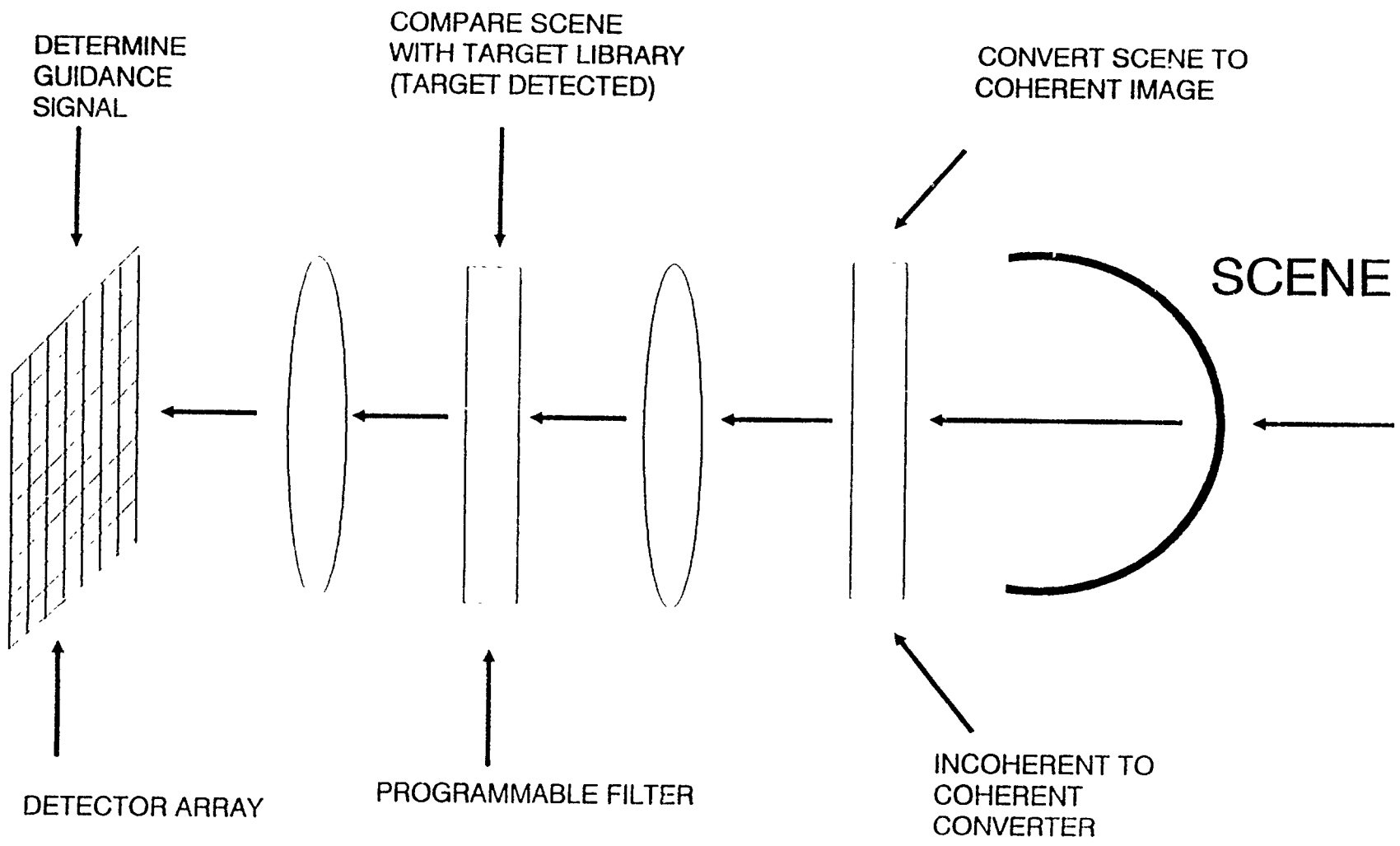
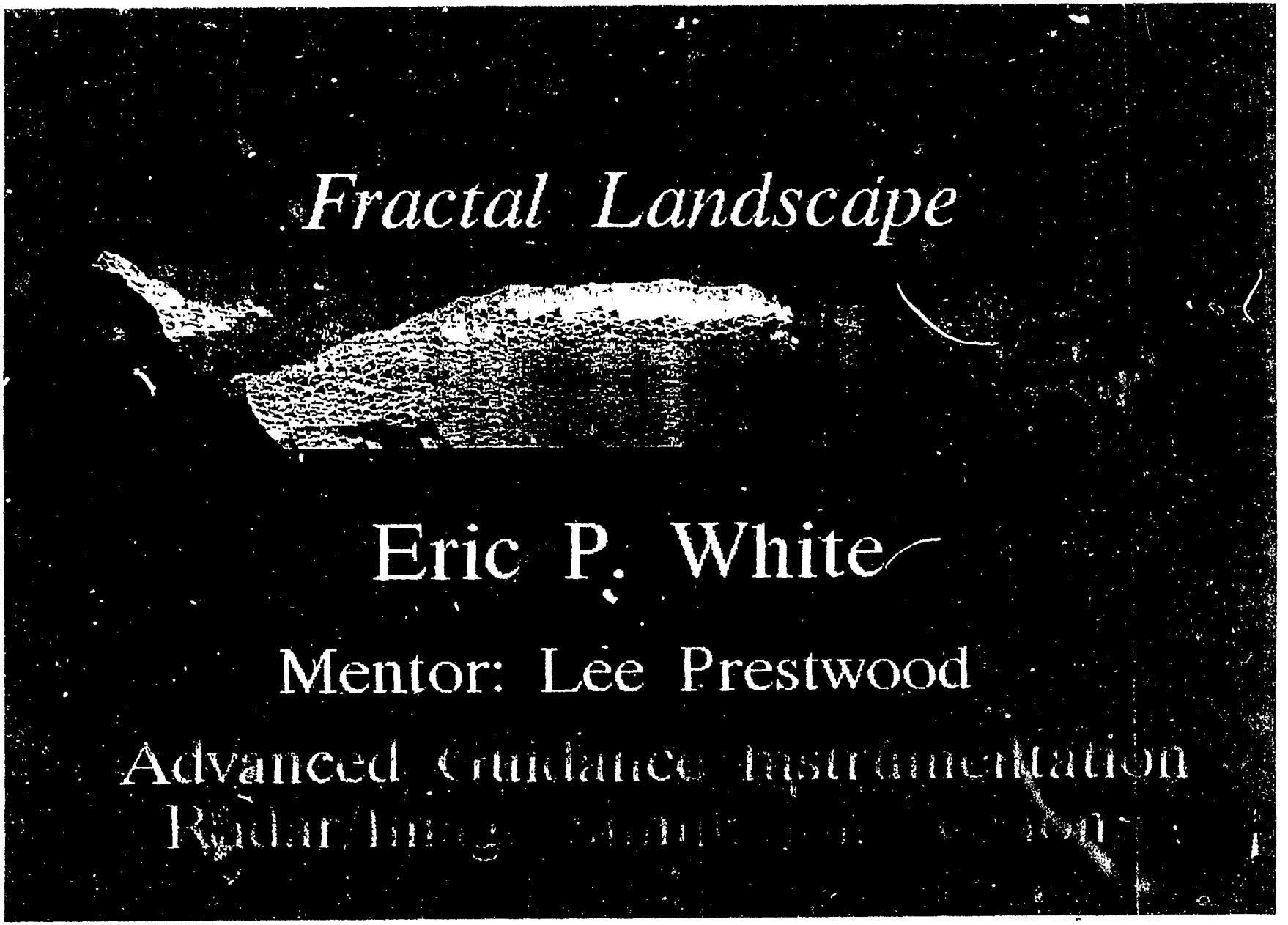


FIGURE 14

Appendices can be obtained from
UNIVERSAL ENERGY SYSTEMS, INC.


Fractal Landscape
Eric P. White
Mentor: Lee Prestwood
Advanced Guidance Instrumentation

Outline

1.0 Introduction

2.0 Acknowledgements

3.0 Background

4.0 Research

  4.1 Fractal Concepts

  4.2 Midpoint Displacement

  4.3 Landscape Programs

5.0 Discussion

  5.1 Advantages of Synthetic Scenery

  5.2 Future Applications

6.0 Experience Gained This Summer

7.0 Program Source Code

8.0 Bibliography

Introduction

Working in an environment such as the Air Force Armament Laboratory provides a learning experience without equal. The section I worked under this summer, Advanced Guidance Instrumentation / Radar and Image Simulation Section (AGI/RISS), helped enhance many of my abilities and gave me the opportunity to learn many new skills. This summer I researched the feasibility of using synthetic terrain models for computer simulations of weapons sensors/seekers.

Acknowledgements

My mentor, Lee Prestwood, and section chief, Mike Wallace, were invaluable to my research this summer. Recognition must also be extended to Thomas Marler, Russell Dukes, Neville Thompson, Chris Ellis, and everyone at AGI for their help and support throughout the summer. And, of course, I must give special thanks to my fellow high school apprentices for supporting me and making my term here more enjoyable.

Background

Realistic landscapes and meaningful background scenery for weapon seeker simulations and tests on computers can be expensive and slow to obtain. This summer I was tasked to research the feasibility of using Iterative Fractal Algorithms (IFA) to

produce synthetic scenery for said simulations. This research started with a review of a few sample computer programs and search of pertinent magazine articles. By the end of my summer session I had successfully modified an IFA to run on a Z-248 personal computer under the Turbo C language.

Research

My research this summer started with the basics of fractal geometry concepts. I studied first the work of B. Mandelbrot's fractal geometry and his Mandelbrot Set (Fig.1). The algorithms and techniques used to produce these scenes are extremely slow, but incredibly simple. Plotting the Mandelbrot Set uses a basic formula, $Z = Z^2 + C$, where Z and C are complex numbers. point is run through this equation and plotted with the number of iterations it takes to go out of a set boundary as the intensity value. After many iterations, unique and picturesque patterns form. Transforming these Mandelbrot pictures into three dimensional figures using the point's intensity as its elevation produces a fantasy realm of peaks and plateaus (Fig.2). Unfortunately, scenery of this type lack realism. A more realistic algorithm deals with midpoints.

The midpoint displacement algorithm generates mountainous scenery by altering the elevation of certain points of a figure. We start with a figure such as a square and calculate the midpoints of its sides and the location of its center (Fig.3). The value of these points are then modified by a random amount.

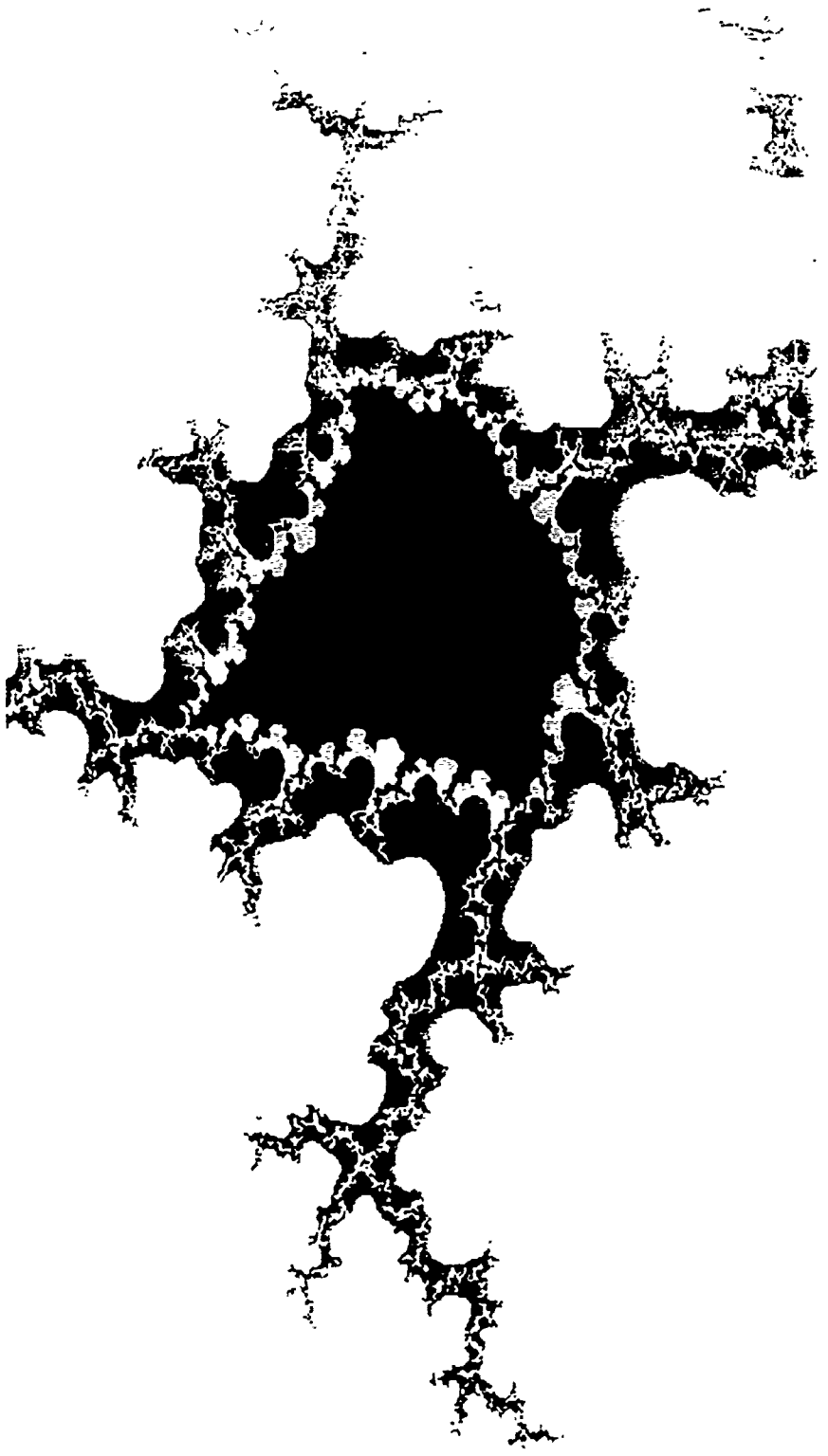


Fig. 1



Fig. 2

The Concept of Midpoint Displacement

- Start with a geometric figure such as a square
- Calculate the midpoint of each segment
- Modify the value of these points by a random amount
- Repeat with newly formed figures as many times as necessary

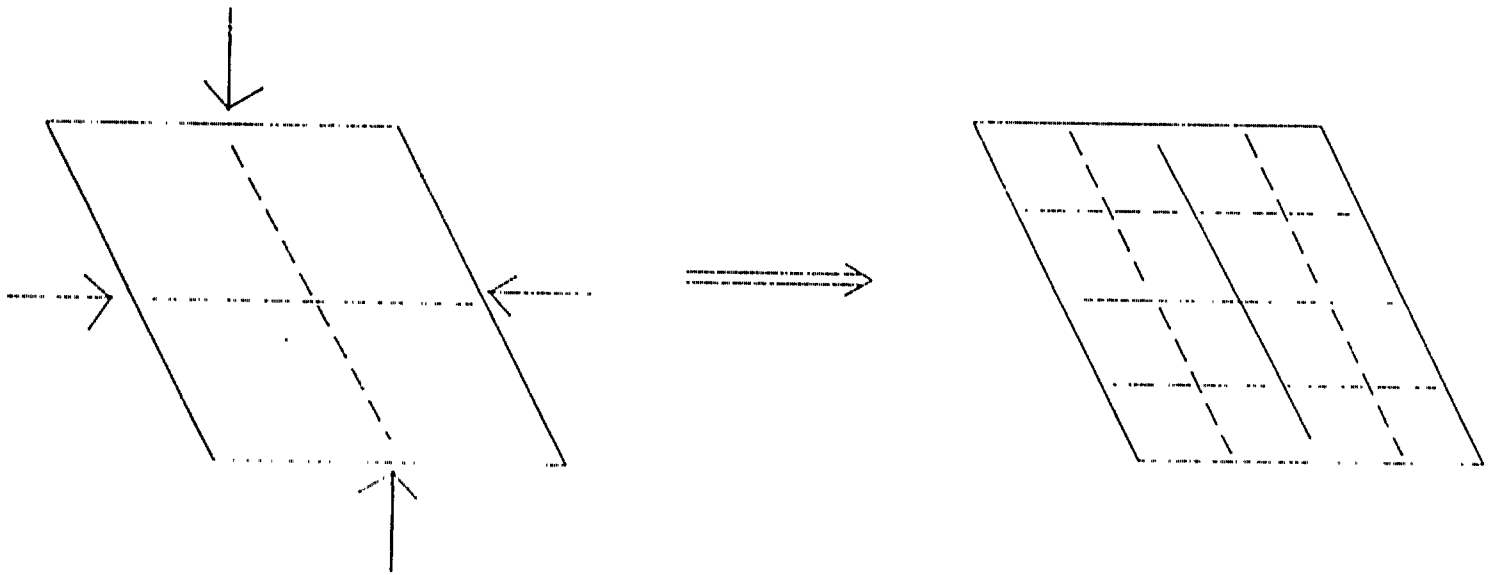


Fig. 3

The figure can then be subdivided into smaller divisions. We continue to repeat the process using each new subdivisions until the scene reaches the desired resolution. This algorithm runs very quickly and produces more realistic results than using the Mandelbrot Set. I have recently read about algorithms that produce even more realistic results. These methods allow for erosion and river patterns. Due to time constraints I could not pursue these methods any further.

The first computer program I executed dealing with fractal landscapes I had found in a magazine. It utilized the midpoint displacement concept that I explained earlier. Unfortunately, the code for the program was written in machine language, and I could not decipher it.

I then turned to another program written in Amiga basic. I used the code structure of this program to create my own program in C. This program worked remarkably well and became the fruit of my efforts. The source code for this program is included at the end of this paper.

Discussion

The use of synthetic terrain for testing simulations holds many significant advantages over conventional methods of target background and imagery acquisition. Methods used most often include photographing actual terrain through radar or actual pictures, and building scale models of the needed scenery. This can be slow and expensive.

Generating landscapes synthetically helps eliminate these problems. IFA programs, like the one I wrote, run very efficiently taking only seconds to generate new terrain. The expense related to generation of this type remains minimal, as well.

In the future, I hope to vary the program's data format for the scenery. Formats such as infra-red, radar, and electro-optical are options that could be implemented at a later date. I also hope to experiment with new algorithms and techniques that create more realistic results.

Experience Gained

Working under this program has given me experience unique to AFATL. In order to complete my research I had to learn the various operating systems and programming languages available on the computers. The RISS hosts two labs that house unique video equipment and powerful mainframe computers. I first had to learn VAX/VMS to operate the VAX mainframes in the labs. I then began learning the languages C and Fortran. In addition to my project, I did some independent research which included learning the UNIX operating system and various utilities in VMS.

```
/*********************************************************/
/***         Fractal landscapes generator              ***/
/*                                                       */
/*    Adapted from an algorithm originally               */
/*    written in Amiga Basic.                            */
/*                                                       */
/*    Modified by Eric P. White, HSAP 1990               */
/*    Mentor: Lee Prestwood AFATL/AGI                    */
/*********************************************************/

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <graphics.h>

double  elevation[65][65], max_elevation = 1, min_elevation = 1;

int scale (double);
void make_mountain (float);
void drawmountain();

main ()
{
  int num=0, num2=0;
  long scaled_elevation;
  unsigned seed;
  float maxvar=3;
  char answer;

/*  printf("\nEnter filename to output data: ");
  gets(filename);*/

  while ((maxvar < 0)||(maxvar > 2))
  {
        printf("\nEnter the maximum variation for the elevation (0-2) ");
        scanf("%f", &maxvar);
  }

  printf("\nEnter a random seed ('1' for default starting point)");
  scanf(" %u", &seed);
  srand(seed);

  for (num = 0; num < 64; num++)
  {
        for (num2 = 0; num2 < 64; num2++)
        {
                elevation[num][num2] = 0.0;
        }
  }


  make_mountain (maxvar);
  drawmountain();

}

void make_mountain (float max)
/*** routine to load randomized values into an array
        based on the neighboring values ***/
```

```
{ int iter, side, x, y;   /*  declares variables for iterations and  */
                          /*  array coordinates  */
  double old, ran, random_value;  /*variables for the random generation*/

  int half_length;  /*variable for half the length of a side of the square*/

  div_t result;

   old = 0; iter=0; side=0; x=0; y=0;
   x = 0;
   for (iter = 6; iter > 0; iter--){  /** main loop **/
        side = pow(2, iter);  /* side is defined as 2 to the power of the
                                 current iteration */
        result = div (side, 2);
        half_length = result.quot;
        printf ("\nDoing iteration %d \n",iter);
/* do the tops and bottoms */
        printf ("Tops & Bottoms \n");
        x=0; y=0;
        for (y = 0; y < 64; y += side)
        {   /* loop for y value of array */
            for (x = half_length; x < 64; x += side)
            {   /* loop for x value */
                ran = (rand()/(32767.0))-.5;        /** calculates the   */
                random_value = ran * max * side;  /*  random value   **/
                old = (elevation[x-half_length][y]) +
                      (elevation[x+half_length][y]);
                old = old/2;  /* allows new value to be proportional to old  */
                elevation[x][y] = old + random_value; /* sets new elevation */
             }
        }
/* do the sides */
        printf ("Sides \n");
        x=0; y=0;
        for (x = 0; x < 64; x += side)
        {   /* loop for y value of array */
            for (y = half_length; y < 64; y += side)
            {   /* loop for x value */
                ran = (rand()/32767.0)-.5;        /** calculates the   */
                random_value = ran * max * side;  /*  random value   **/
                old = (elevation[x][y-half_length]) +
                      (elevation[x][y+half_length]);
                old = old/2;  /* allows new value to be proportional to old  */
                elevation[x][y] = old + random_value; /* sets new elevation */
             }
        }
/* do the centers of the squares */
        printf ("Centers \n");
        x=0; y=0;
        for (x = half_length; x < 64; x += side)
        {   /* loop for x value of array */
            for (y = half_length; y < 64; y += side)
            {   /* loop for y value */
                ran = (rand()/32767.0)-.5;        /** calculates the   */
                random_value = ran * max * side;  /*  random value   **/
                old = (elevation[x-half_length][y+half_length]) +
                      (elevation[x+half_length][y-half_length]) +
                      (elevation[x+half_length][y+half_length]) +
                      (elevation[x-half_length][y-half_length]);
                old = old/4;  /* averages 4 values instead of 2  */
```

```
                elevation[x][y] = old + random_value; /* sets new elevation */
                if (elevation[x][y] > max_elevation)
                        max_elevation = elevation[x][y];
                if (elevation[x][y] < min_elevation)
                        min_elevation = elevation [x][y];
            }
        }
    }
}

int scale (double value)
{
  double new_elevation;
  value = (value)/ (max_elevation);
  new_elevation = value*255.0;
  return (new_elevation);
}


void drawmountain()

{
 float xshift=1.5, xm=7.0, ym=1.5, yp=100.0, waterline=0.0, snowline;
 int a, b, x, y, num, num2;
 int height, rx1, ry1, rx2, ry2, rx3, ry3, rx4, ry4, ry, rx;
 int graphdriver = DETECT, graphmode, numpt = 3,
     triangle[6], errorcode;

 /* Detect adapter type and initialize graphics system */
/* if (registerbgidriver(EGAVGA_driver) < 0)
 {
    printf("Driver could not be registered!\n");
    exit(0);
 }*/
 initgraph(&graphdriver, &graphmode, "c:\\tc");
 errorcode = graphresult();
 if (errorcode != grOk)                {
   printf("Graphics error: %s\n",
   grapherrormsg(errorcode));
   exit(1);
 };

  snowline = max_elevation - (max_elevation/4);

  for (num = 0; num < 64; num++)
  {
        for (num2 = 0; num2 < 64; num2++)
        {
                if (elevation[num][num2] < 0.0) elevation[num][num2] = 0.0;
        }
  };

 for (x = 63; x >= 0; x--)
 {
  for (y = 63; y >= 0; y--)
  {
   height = (elevation[x][y])+
            (elevation[x+1][y])+
            (elevation[x][y+1])+
            (elevation[x+1][y+1]);
```

```
height = height/4;
a = x; b = y;
rx1 = (xm * a) + (xshift * b);
ry1 = (ym * b) + yp - elevation[a][b];
/* get a shade later*/
a = (x + 1);
rx2 = (xm * a) + (xshift * b);
ry2 = (ym * b) + yp - elevation[a][b];
/* get a shade later*/
a = x; b = (y + 1);
rx3 = (xm * a) + (xshift * b);
ry3 = (ym * b) + yp - elevation[a][b];
/* get a shade later*/
a = (x + 1);
rx4 = (xm * a) + (xshift * b);
ry4 = (ym * b) + yp - elevation[a][b];
/*get shade later*/
a = (x + .5); b = (y + .5);
rx = (xm * a) + (xshift * b);
ry = (ym * b) + yp;
a = x; b = y;
ry = ry - height;
  triangle[0] = rx;
  triangle[1] = ry;
  triangle[2] = rx1;
  triangle[3] = ry1;
  triangle[4] = rx2;
  triangle[5] = ry2;
if (elevation[a][b] <= waterline)
{
   setfillstyle (SOLID_FILL, LIGHTBLUE);
   setcolor (LIGHTBLUE);
}
else
{
   setfillstyle (SOLID_FILL, GREEN);
   setcolor (LIGHTGREEN);
}
if (elevation[a][b] > snowline)
{
   setfillstyle (SOLID_FILL, LIGHTGRAY);
   setcolor (WHITE);
}
fillpoly(numpt, triangle);
  triangle[0] = rx;
  triangle[1] = ry;
  triangle[2] = rx2;
  triangle[3] = ry2;
  triangle[4] = rx4;
  triangle[5] = ry4;
if (elevation[a][b] <= waterline)
{
   setfillstyle (SOLID_FILL, LIGHTBLUE);
   setcolor (LIGHTBLUE);
}
else
{
   setfillstyle (SOLID_FILL, GREEN);
   setcolor (LIGHTGREEN);
}
```

```
      if (elevation[a][b] > snowline)
      {
         setfillstyle (SOLID_FILL, LIGHTGRAY);
         setcolor (WHITE);
      }
      fillpoly(numpt, triangle);
        triangle[0] = rx;
        triangle[1] = ry;
        triangle[2] = rx1;
        triangle[3] = ry1;
        triangle[4] = rx3;
        triangle[5] = ry3;
      if (elevation[a][b] <= waterline)
      {
         setfillstyle (SOLID_FILL, LIGHTBLUE);
         setcolor (LIGHTBLUE);
      }
      else
      {
         setfillstyle (SOLID_FILL, GREEN);
         setcolor (LIGHTGREEN);
      }
      if (elevation[a][b] > snowline)
      {
         setfillstyle (SOLID_FILL, LIGHTGRAY);
         setcolor (WHITE);
      }
      fillpoly(numpt, triangle);
        triangle[0] = rx;
        triangle[1] = ry;
        triangle[2] = rx3;
        triangle[3] = ry3;
        triangle[4] = rx4;
        triangle[5] = ry4;
      if (elevation[a][b] <= waterline)
      {
         setfillstyle (SOLID_FILL, LIGHTBLUE);
         setcolor (LIGHTBLUE);
      }
      else
      {
         setfillstyle (SOLID_FILL, GREEN);
         setcolor (LIGHTGREEN);
      }
      if (elevation[a][b] > snowline)
      {
         setfillstyle (SOLID_FILL, LIGHTGRAY);
         setcolor (WHITE);
      }
      fillpoly(numpt, triangle);
    }
  }
/*   outtextxy(10, getmaxy()-30," Press any key");*/
   getch();
   closegraph();
}
```

Bibliography

Barkakati. The Waite Group's Turbo C Bible. Howard W. Sams and Company, Indianapolis. 1989.

Dewdney, A.K. "Computer Recreations". Scientific American. Feb., 1989. pg. 108-111.

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

Peitgen, Heinz-Otto and Saupe Dietmar. The Science of Fractal Images. Springer-Verlag, New York. 1988.

Timmerman, Matthew. "Fractal Mountains for Amiga". Compute!. Aug., 1987. pg. 89-91.