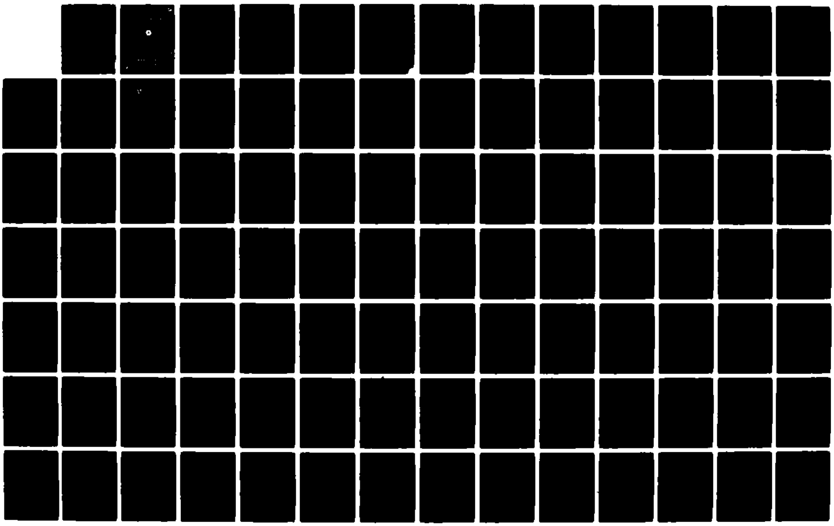
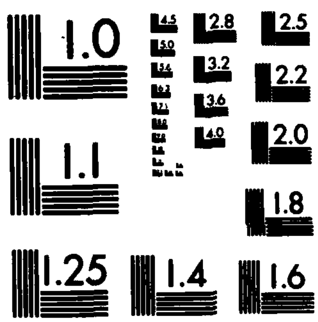
AD-A124 343 HHE/LORAN-C SURVEYING(U) COAST GUARD WASHINGTON DC OFFICE OF RESEARCH AND DEVELOPMENT A J SEDLOCK NOV 82 USCG-D-54-82 1/4

UNCLASSIFIED F/G 17/7 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Report No. CG-D-54-82

HHE/LORAN-C SURVEYING

LCDR A.J. SEDLOCK, USCG

NOVEMBER 1982

FINAL REPORT

Document is available to the public through the
National Technical Information Service,
Springfield, Virginia 22161

ADA124343

DTIC
FEB 1 4 1983

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
United States Coast Guard
Office of Research and Development
Washington, D.C. 20590

DTIC FILE COPY

83 02 11 036

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The contents of this report do not necessarily reflect the official view or policy of the Coast Guard; and they do not constitute a standard, specification, or regulation.

This report, or portions thereof may *not* be used for advertising or sales promotion purposes. Citation of trade names and manufacturers does not constitute endorsement or approval of such products.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
CG-D-54-82	A124342	

4. Title and Subtitle	5. Report Date
HHE/LORAN-C SURVEYING	NOVEMBER 1982
	6. Performing Organization Code: G-DST-1
	8. Performing Organization Report No.

7. Author(s)	
LCDR A. J. SEDLOCK	

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
COMMANDANT (G-DST-1) U. S. COAST GUARD 2100 SECOND ST. SW WASHINGTON, DC 20593	2123
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A methodology ~~has been~~ was developed to provide calibration data for Loran-C navigators designed to be used in harbor, harbor entrance areas. The methodology can be applied over a wide range of local environments. Visual aids to navigation are used as a position reference whenever possible. An electronic positioning system is used when available visual aids to naviagation are not satisfactory. The data acquisition system, data collection techniques and data analysis procedures have been field tested in New York Harbor, Delaware Bay and River and the St. Marys River (Michigan).

The calibration points, waypoints, are calculated for each intersection of commonly used tracklines in a harbor area. Additional calibration points, termed trackpoints, are added as necessary to minimize position errors.

A typical harbor with 20-30 waypoints can be surveyed over a two week period. A survey crew consists of three to five persons; a survey using only visual aids could be accomplished with as few as two. The survey data is analyzed in the field as the survey progresses to prevent the need to resurvey. Troublesome data is identified, and ~~the data is recollected~~.

This document provides a brief description of the data acquisition system, Time-Difference Survey System (TDSS); and a detailed description of data collection and data analysis procedures. (A complete description of the TDSS may be found in the Coast Guard Research and Development Center Report, "Time Difference Survey System (TDSS)"). Improvements and expanded utilization to the TDSS is discussed. Also discussed is a technique analogous to a track plotter for user equipment self-calibration.

17. Key Words	18. Distribution Statement
Loran-C, Precision Loran-C, Harbor Navigation, Piloting, Precision Harbor Navigation, Surveying, Grid Surveying, Grid Calibration, Loran-C Surveying	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED		

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

METRIC CONVERSION FACTORS

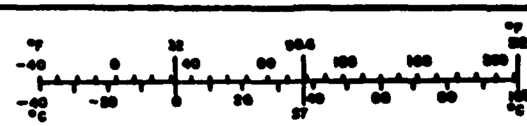
Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
	AREA			
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
	MASS (weight)			
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
	VOLUME			
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
	TEMPERATURE (exact)			
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
	AREA			
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
	MASS (weight)			
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
	VOLUME			
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
	TEMPERATURE (exact)			
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F

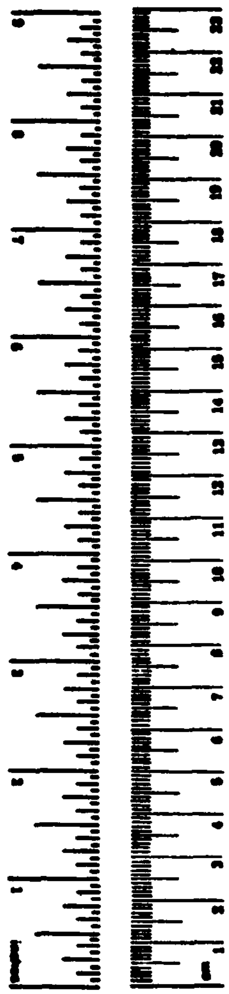


METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

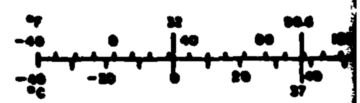
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from M

Symbol	When You Know	Multiply by
		LENGTH
mm	millimeters	0.04
cm	centimeters	0.4
m	meters	3.3
m	meters	1.1
km	kilometers	0.6
		AREA
cm^2	square centimeters	0.16
m^2	square meters	1.2
km^2	square kilometers	0.4
ha	hectares (10,000 m^2)	2.5
		MASS (weight)
g	grams	0.035
kg	kilograms	2.2
t	tonnes (1000 kg)	1.1
		VOLUME
ml	milliliters	0.03
l	liters	2.1
l	liters	1.06
l	liters	0.26
m^3	cubic meters	35
m^3	cubic meters	1.3
		TEMPERATURE (exa
°C	Celsius temperature	9/5 (then add 32)

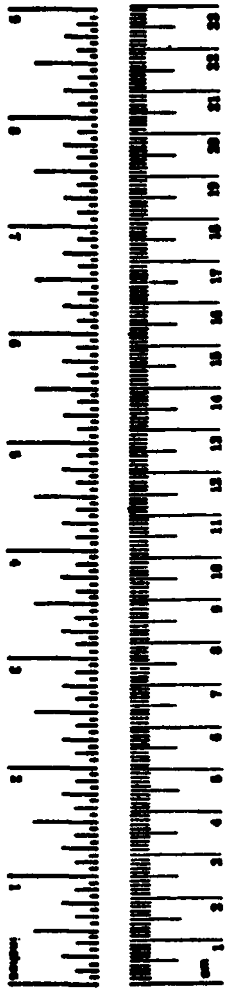


METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

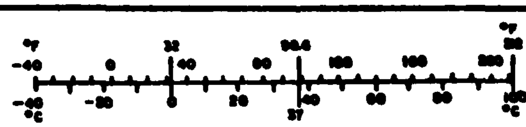
When You Know	Multiply by	To Find	Symbol
LENGTH			
inches	*2.5	centimeters	cm
feet	30	centimeters	cm
yards	0.9	meters	m
miles	1.6	kilometers	km
AREA			
square inches	6.5	square centimeters	cm^2
square feet	0.09	square meters	m^2
square yards	0.8	square meters	m^2
square miles	2.6	square kilometers	km^2
acres	0.4	hectares	ha
MASS (weight)			
ounces	28	grams	g
pounds	0.45	kilograms	kg
short tons (2000 lb)	0.9	tonnes	t
VOLUME			
teaspoons	5	milliliters	ml
tablespoons	15	milliliters	ml
fluid ounces	30	milliliters	ml
cups	0.24	liters	l
pints	0.47	liters	l
quarts	0.95	liters	l
gallons	3.8	liters	l
cubic feet	0.03	cubic meters	m^3
cubic yards	0.76	cubic meters	m^3
TEMPERATURE (exact)			
Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
	AREA			
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
	MASS (weight)			
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
	VOLUME			
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
	TEMPERATURE (exact)			
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



DTIC COPY INSPECTED 3

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A

HHE/LORAN-C SURVEYING

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

SUMMARY

A methodology has been developed to provide calibration data for Loran-C navigators designed to be used in harbor, harbor entrance areas. The methodology can be applied over a wide range of local environments. Visual aids to navigation are used as a position reference whenever possible. An electronic positioning system is used when available visual aids to navigation are not satisfactory. The data acquisition system, data collection techniques, and data analysis procedures have been field tested in New York Harbor, the Delaware Bay and River, and the St. Marys River.

The calibration points, waypoints, are calculated for each intersection of commonly used tracklines in a harbor area. Additional calibration points, termed trackpoints, are added as necessary to minimize position errors.

A typical harbor with 20-30 waypoints can be surveyed over a two week period. A survey crew consists of three to five persons; a survey using only visual aids could be accomplished with as few as two. The survey data is analyzed in the field as the survey progresses to prevent the need to resurvey. Troublesome data is identified, and the data is recollected.

This document provides a brief description of the data acquisition system, Time-Difference Survey System (TDSS); and a detailed description of data collection and data analysis procedures. (A complete description of the TDSS can be found in the Coast Guard Research and Development Center Report, "Time Difference Survey System (TDSS)"). Improvements and expanded utilization to the TDSS is discussed. Also discussed is a technique analogous to a track plotter for user equipment self-calibration.

INTRODUCTION

Loran-C harbor survey techniques have been developed to provide calibration data for Loran-C navigators designed to be used in harbors and harbor-entrance (HHE) areas. Calibration points are provided for each straight line channel intersection, i.e., a waypoint. A calibration point between waypoints termed a trackpoint can be added, if necessary, to reduce position errors to an acceptable level (e.g., less than 10 meters cross-track) in areas where there is grid warp. Included in the survey techniques are procedures to validate the waypoints and detect the need for trackpoints. A data acquisition system for collecting Loran-C time-difference (TD) data and electronic position data has been designed, implemented and field tested. The survey techniques are designed such that the data is collected and waypoints are calculated and verified in the field.

Two general methods of surveying waypoints have been developed. One method is used in restricted waterways where it is impractical to use a short range electronic positioning system. These waterways must be marked well by visual aids to navigation (particularly visual range markers) and have distinct channel boundaries. A good example of such a waterway is the St. Marys River, which connects Lake Superior and Lake Huron. The second method is designed for areas where a short-range electronic positioning system is practical. In this case there are no requirements for visual ranges and distinct channel boundaries. A good example of such an area is outer New York harbor. The first method is termed Visual Reference Survey; the second, Electronic Positioning Augmentation.

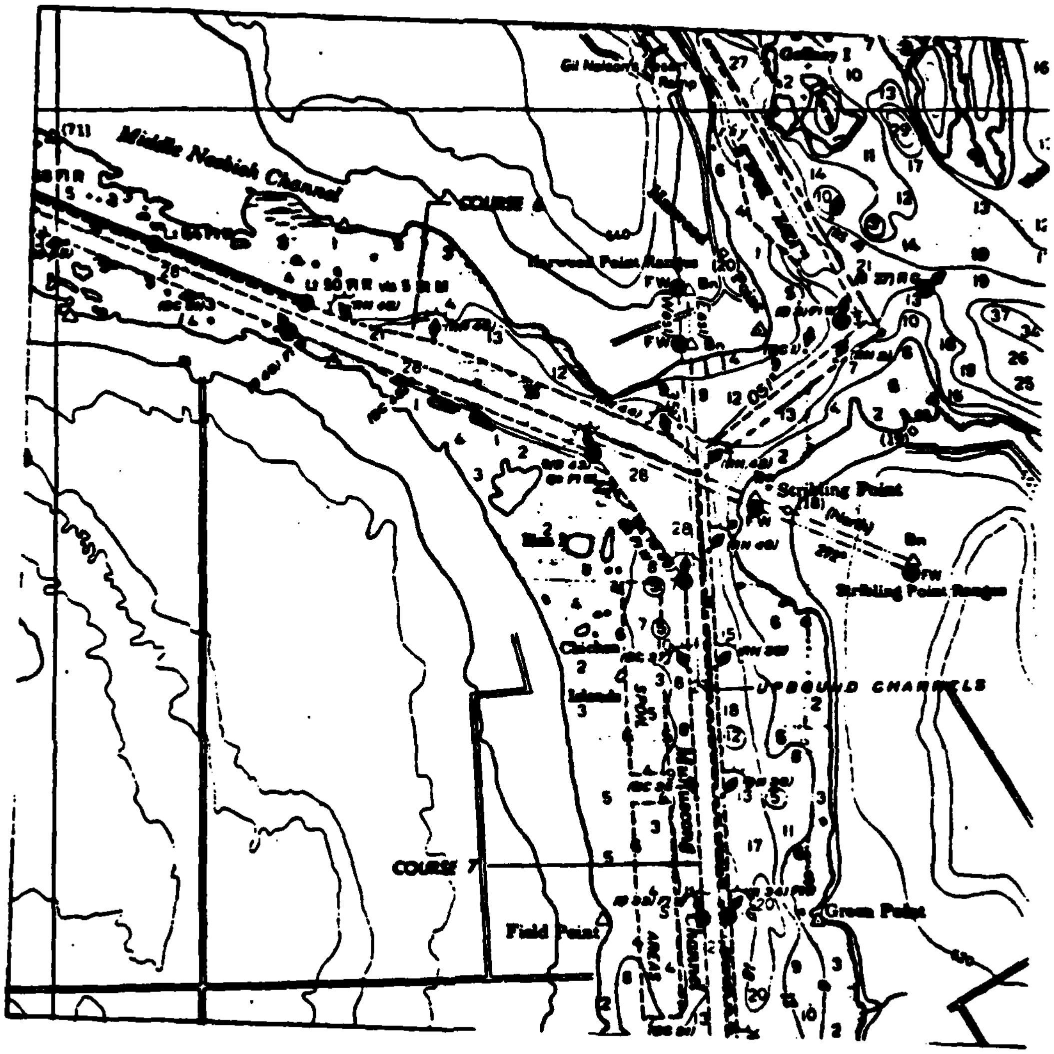
VISUAL REFERENCE SURVEY

SURVEY APPROACHES

In areas such as a winding river it is an expensive and time consuming operation to setup and operate a short range electronic positioning system (e.g. Mini-Ranger, Autotape, Etc.) to provide a position reference for a Loran-C TD waypoint survey. Two effective techniques have been developed which utilize existing visual aids to navigation as the positioning reference. The first technique, Range-Range, is designed for the case where both channels which define the waypoint are marked by visual ranges. Examples of such a case are shown in figures 1A and 1B. The second technique, Channel-Edge, is designed for the case where one or both channels defining the waypoint is not marked by a range, but the channel edges are well defined (easily detectable by a fathometer) and/or marked by fixed and floating aids to navigation. An example of such a situation is shown in figure 2.

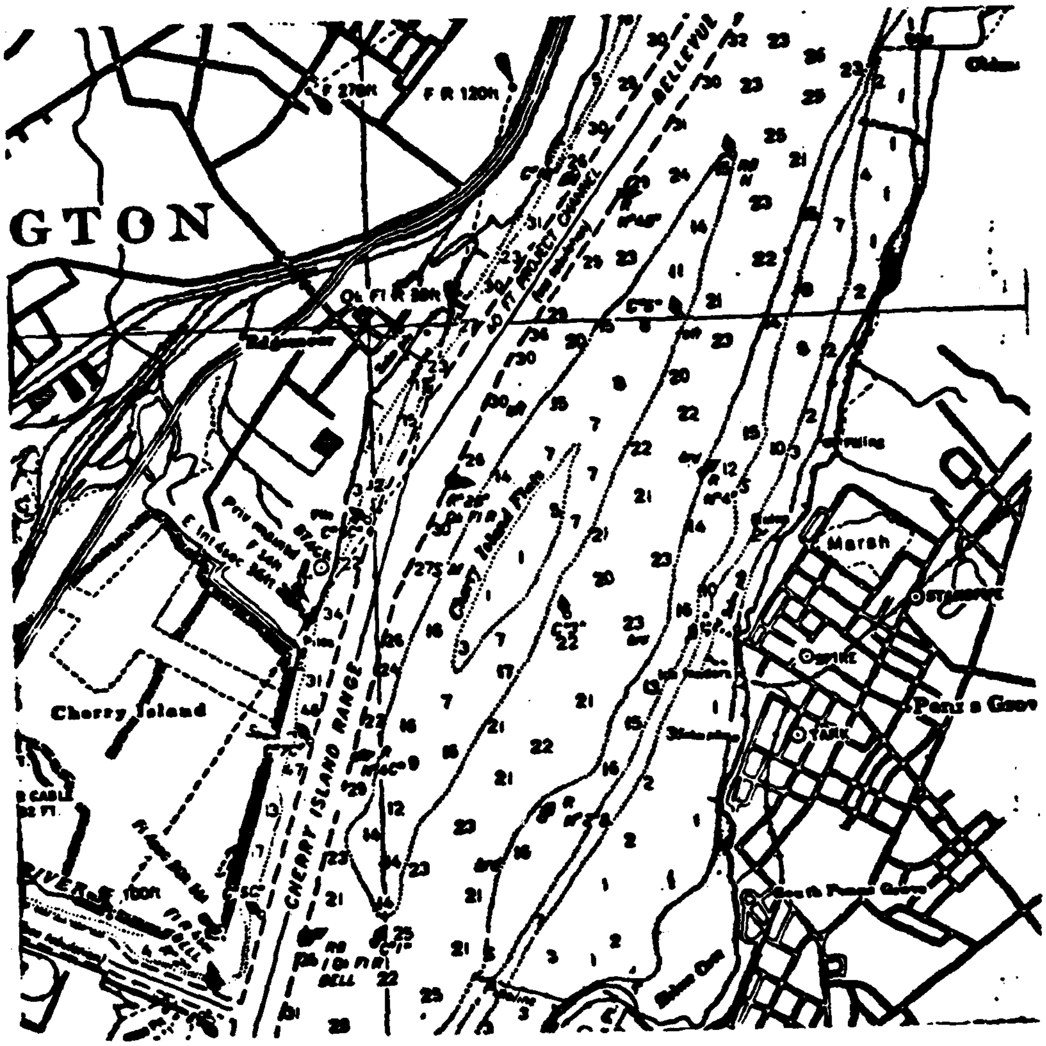
Range-Range Survey

In the case where both channels which define the waypoint are marked by visual ranges, the waypoint is defined as the point where both ranges close simultaneously. The visual ranges provide an excellent cross track reference near the channel centerline. An attempt could be made to survey the waypoint by maneuvering the survey vessel until both ranges are closed and then attempting to maintain this station until sufficient Loran TD data is collected. Unfortunately, such a technique is impractical to implement due to the wind, current and other vessel traffic conditions normally encountered.



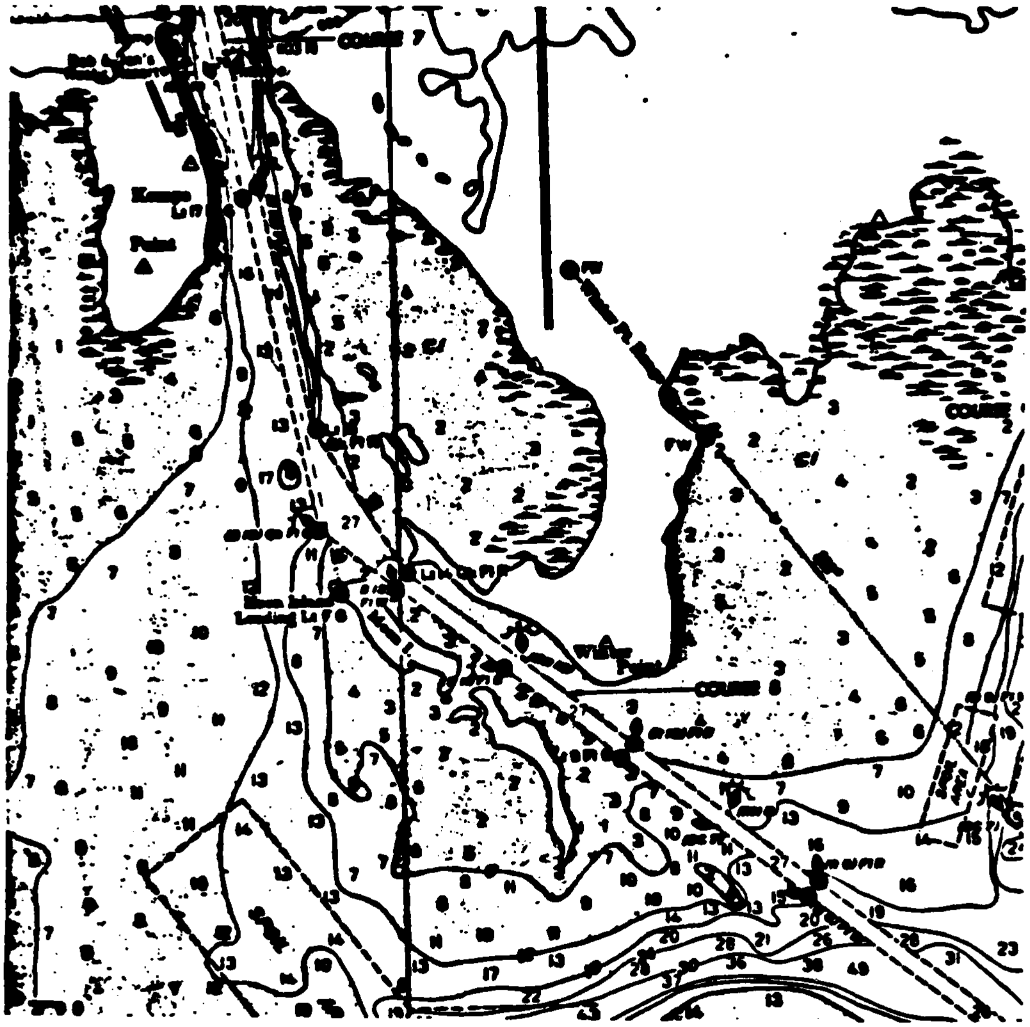
WAYPOINT DEFINED BY TWO VISUAL RANGES (ST. MARYS RIVER)

FIGURE 1A



WAYPOINT DEFINED BY TWO VISUAL RANGES (DELEWARE RIVER)

FIGURE 1B



WAYPOINT FOR CHANNEL EDGE SURVEY (ST. MARYS RIVER)

FIGURE 2

An alternative approach is described below.

Mathematical Basis

In the area about a waypoint, the Loran-C TD lines-of-position (LOPs) can be approximated by the linear model below:

$$TDX-TDX_o = a_{11}(x-x_o) + a_{12}(y-y_o) + n_x \quad (1)$$
$$TDY-TDY_o = a_{21}(x-x_o) + a_{22}(y-y_o) + n_y \quad (2)$$
$$TDZ-TDZ_o = a_{31}(x-x_o) + a_{32}(y-y_o) + n_z \quad (3)$$

where

TDX, TDY, TDZ are observed TDs
TDX_o, TDY_o, TDZ_o are waypoint TDs
x, y are position coordinates of the observation
x_o, y_o are waypoint position coordinates of the waypoint
$a_{i,j}$ are coefficients of the gradient matrix (directional derivatives of the TD grid)
n_x, n_y, n_z are error terms due to noise and nonlinearities

Position and TD data collected during an experiment in the Groton, CT, area displayed excellent agreement (i.e., rms errors less than .050 microsec) with the above model within 1.5 km of a waypoint.

The equations for the centerline of the channels defining the waypoint can be written in the following form.

$$(y-y_o) = M_1\ (x-x_o) \quad (4)$$
$$(y-y_o) = M_2\ (x-x_o) \quad (5)$$

where y = north-south position
x = east-west position
M_1,M_2 = tan (course lines)
x_o,y_o - waypoint coordinates

The equations for the channel centerlines can be expressed totally in Loran-C TD coordinates by substituting equations (4) and (5) into equations (1), (2) and (3). For example:

$$\text{Trackline 1:}\quad (TDY-TDY_o) = b_1(TDX-TDX_o) + N_1 \quad (6)$$
$$\text{Trackline 2:}\quad (TDY-TDY_o) = b_2(TDX-TDX_o) + N_2 \quad (7)$$

where
$$b_1 = (a_{21} + a_{22}M_1)/(a_{11} + a_{12}M_1) \quad (8)$$
$$b_2 = (a_{21} + a_{22}M_2)/(a_{11} + a_{12}M_2) \quad (9)$$

$$N_1 = n_y - b_1 n_x \qquad (10)$$
$$N_2 = n_y - b_2 n_x \qquad (11)$$

The equations for the tracklines can be similarly expressed in terms of TDX and TDZ and TDY and TDZ.

Loran-C TD data can be collected along the centerlines of each channel near the waypoint using the visual ranges. An estimate of the trackline equations in TD coordinates can be calculated using linear regressions to the data. These equations are in a slightly different form than equations (6) and (7).

trackline 1 regression line: $TDY-TDY_1=C_1(TDX-TDX_1)$ (12)
trackline 2 regression line: $TDY-TDY_2=C_2(TDX-TDX_2)$ (13)

where

TDX_1, TDX_2 = mean of the TDX data collected on trackline 1, 2
TDY_1, TDY_2 = mean of the TDY data collected on trackline 1, 2
C_1 = calculated slope from the linear regression of trackline 1 data
C_2 = calculated slope from the linear regression of trackline 2 data
Slope = r(Stdy/Stdx) (14)

Stdy = standard deviation of TDY data
Stdx = standard deviation of TDX data
r = correlation coefficient of TDX and TDY data

Waypoint Estimation

An estimate of the waypoint TDs (TDX_o, TDY_o) is determined by calculating the intersection of equations (12) and (13).

For each pair of Loran-C TDs collected, a set of regression lines and their intersection is calculated. If three TDs are collected, there are two estimates calculated for each waypoint TD (e.g., TDX_o and TDY_o estimated from TDX/TDY regression lines, TDX_o and TDZ_o from TDX/TDZ regression lines, and TDY_o and TDZ_o estimates from TDY/TDZ regression lines). If four TDs are collected, three estimates for each waypoint TD are calculated.

The accuracy of the waypoint estimates is a function of:

a. the crossing angle of the tracklines (in TD coordinates)

b. the confidence bounds for the regression lines at the point of intersection.

$$\text{estimated rms error of waypoint} = (S1^2 + S2^2)^{1/2}/\sin(A) \tag{15}$$

where:

S1 = one sigma confidence bound of trackline one regression line at the calculated waypoint

S2 = one sigma confidence bound of trackline two regression line at the calculated waypoint

A = crossing angle of the tracklines for equations (12) and (13) in time difference coordinates

$$A = ABS(\arctan(C_2) - \arctan(C_1)) \tag{16}$$

$$S1^2 = RES1^2\ (1/N1 + (TDX_o - TDX1)^2/((N1-2)(Stdx1^2))) \tag{17}$$

$$S2^2 = RES2^2\ (1/N2 + (TDX_o - TDX2)^2/((N2-2)(Stdx2^2))) \tag{18}$$

N1 = Number of data samples on trackline one
N2 = Number of data samples on trackline two

RES1,RES2 = Standard deviation of residuals for the regression line fit to trackline one data, trackline two

Stdx1,Stdx2 = Standard deviation of TDX data for trackline one data, trackline two
Stdy1,Stdy2 = Standard deviation of TDY data for trackline one data, trackline two

$$RES1^2 = Stdy1\ (1-r_1)^2 \tag{19}$$

$$RES2^2 = Stdy2\ (1-r_2)^2 \tag{20}$$

r1,r2 = correlation coeffecient of TDX and TDY for trackline one data, trackline two

The crossing angles of the tracklines in TD coordinates are determined by the orientation of the tracklines (in spatial coordinates) and the locations of the Loran-C transmitters. Control of the accuracy of the estimated waypoint coordinates depends on the ability to minimize the confidence bounds of the regression lines at the waypoint. From equations (16) and (17) the confidence bounds for the regression lines at the waypoint are functions of:

a. the residuals (RES) for the regression line fit to the data. The residual is the standard deviation of the difference between actual data points and the regression line equation. The smaller the residuals, the better the fit of the line to the data. The residuals of the trackline are a function of the signal-to-noise ratio of the TDs measured and the track keeping of the survey vessel. The trackline length should be long enough such that length of the trackline in microseconds is much greater than the standard deviation of the TDs due to noise. A good rule of thumb is that the trackline length be at least 35 times the standard deviation of the TDs at dockside (e.g., 1-1.5 microseconds).

b. the number of data samples (N). The confidence bound decreases by the square root of the number of data samples. Thus, there is diminishing return to taking more than 100 samples.

c. the distance (in microsec) between the mean of the data set and the calculated waypoint, (TDX_o-TDX). Ideally, the survey pattern should form an X. The worst case is when the survey pattern forms a V. If the survey pattern forms a V, the (TDX_o-TDX) term is maximum. (The survey lines for a V should always at least reach the waypoint.) Assuming data is collected uniformly along the survey trackline and the survey line stops at the waypoint, the term:

$$(TDXo-TDXi)^2/Stdxi = 3 \tag{21}$$

To achieve the same accuracy figure with a V survey as an X survey with the same residuals R:

$$R^2(1/Nx) = R^2(1/Nv + 3/Nv) \tag{22}$$

$$Nv = 4Nx \tag{23}$$

The number of samples in the V survey (Nv) must be four times the number of samples in the X survey(Nx).

Channel Edge Survey

In the case where the one or both of the channels defining the waypoint is not marked by a visual range, the TDs of the waypoint can be calculated without resorting to an electronic positioning system if there are distinct channel boundaries and/or good visual aids to navigation. Loran-C TD data can be collected along the channel edges and at aids to navigation near the waypoint. The difference in TDs between an aid-to-navigation (preferably fixed) and the waypoint are calculated based on the relative positions of the aid from the waypoint. The difference in position establishes the offsets to be applied to the TDs measured at the aid-to-navigation to form an estimate of the waypoint TDs. The position offset between the aid and the waypoint can be determined from the local navigation chart or Army Corps of Engineers (COE) dredging data. The estimated waypoint TDs can be used as a reference point for converting the TDs collected along channel edges and at aids-to-navigation to position coordinates. The position data can be plotted to the chart scale to produce an overlay to the navigation chart or converted to along/cross track and plotted with respect to the channel centerlines. The plots can be compared to navigation chart and/or COE data to detect systematic errors. If an error is detected, a correction to the waypoint TDs must be calculated. The TD data can be again converted to position data referenced to the waypoint. The position data can then be compared to the navigation chart and COE data to verify that the correction was applied correctly. The procedure must be repeated until there are no detectable systematic errors.

Waypoint Estimation

A general expression for calculating Loran-C time difference at a position (x,y) is:

$$TD = (Rs-Rm)/Vp+(SF_s-SF_m)+(ASF_s-ASF_m) - Emis \tag{24}$$

where:

TD = calculated TD

Vp = velocity of propagation corrected for the index of refraction; for index refraction = 1.000338
$1/Vp$ = 2.998986 microsec/kilometer

Rs,Rm = range to secondary station, master station
$Rs = ((Xs-X)^2+(Ys-Y)^2)^{1/2}$

SFs,SFm = correction factor for progation delays over sea water; for Rs and Rm greater than 160 km, SFs-SFm = .0011(Rs-Rm)

ASFs,ASFm = correction factor for additional propagation delay for over land paths

Emis = secondary station emission delay

For harbor survey applications, the terms in equation (24) after (Rs-Rm)/Vp can be treated as a constant for the harbor area, i.e.,

$$TD = (Rs-Rm)/Vp + C \tag{25}$$

The difference in TDs between two points is then:

$$TD2-TD1 = (Rs2-Rm2)/Vp-(Rs1-Rm1)/Vp \tag{26}$$

or

$$TD2 = TD1 + (Rs2-Rm2)/Vp-(Rs1-Rm1)/Vp \tag{27}$$

The above expression can be used to calculate the estimated difference in TDs between an aid-to-navigation (TD1) and a waypoint (TD2).

Aid and Channel Edge Plotting

The algorithm developed for the PILOT Loran-C navigator[1] can be used to convert TD data to position coordinates. This algorithm calculates the differential position on the difference between the measured TDs and the waypoint TDs. This differential position is then added to the waypoint position. The algorithm is basically the inverse of equation (27).

An initial position estimate is made using the linear expression.

$$\underline{Z}_1 = \underline{Z}_o+\underline{G}(\underline{TD-TD_o}) \tag{28}$$

where

$\underline{Z}_1$ = first position estimate
$\underline{Z}_o$ = waypoint position
$\overline{TD}$ = measured TDs
TD_o = waypoint TD
$\underline{G}$ = gradient matrix

The TDs corresponding to Z_1, TD_{Z1}, are calculated using equation (27). A second position estimate, Z_2, is then calculated;

$$\underline{Z}_2 = \underline{Z}_1 + \underline{G}(\underline{TD}-\underline{TD}_{Z1}) \tag{29}$$

If the difference between Z_2 and Z_1 is less than one meter, the calculation is ended. If not, a third position calculation is made:

$$\underline{Z}_3 = \underline{Z}_2 + \underline{G}(\underline{TD}-\underline{TD}_{Z2}) \tag{30}$$

The procedure is repeated until the difference between successive position calculation is less than one meter.

Data to verify a range-range solution is collected along the entire length of the channel centerline using the visual range to maintain the survey vessel on the centerline. If there is a distinct or well marked channel edge, TD data is also collected along channel edges. The TD data is converted to xy and along/cross track positions and compared to the navigation chart and COE data if available. The along/cross track plot of the centerline data will indicate if the waypoint calculations are correct and if there is a requirement for a trackpoint between the waypoints. See the *Data Analysis* section.

Verification

The channel edge survey is self-verifying. The waypoint is chosen such that TD data converted to xy positions agrees with the navigation chart. The solution is normally checked with a redundant set of channel edge data not used in the original waypoint solution.

PRE-SURVEY PLANNING

Overview

The first step in the pre-survey planning process is to determine if the visual survey technique can be applied to the area of interest. In some cases, a portion of the area can be surveyed using visual techniques and the remainder of the area will require electronic positioning augmentation. The remainder of the section assumes that it has been determined that the visual survey is applicable to at least a portion of the area of interest. Presurvey planning for electronic positioning augmentation is discussed under Electronic Positioning Augmentation - Presurvey Planning. The visual survey technique can be applied in areas where the channels are marked by visual ranges and/or where the channels have distinct, well marked boundaries.

During the presurvey planning stages of a Loran-C harbor survey, the navigation charts for the area are studied to define waypoints and select a survey strategy for each waypoint. Some calculations are also performed to assist in the data collection phase.

Waypoint Definition

If the channels defining the waypoint are both marked by visual ranges, the waypoint is chosen as the intersection of the two visual ranges. In some cases the ranges may not mark the centerline of the channel boundaries. Such an example is seen in figure 1A, a section of the St. Marys River. If the channels are not marked by ranges, the intersection of the common tracklines (e.g., channel centerlines) are defined as the waypoints. Local A to N personnel and/or Pilot Association personnel should be consulted if there is any doubt as to where a waypoint should be defined or to determine common vessel tracks. It is important that each waypoint be uniquely defined.

COE Survey Data

Dredging charts and data can be obtained from the COE office responsible for the area of interest. This data provides more detail on channel boundaries, channel width and length, and channel courses than the conventional navigation charts. The COE data is particularly useful for the channel edge survey technique. Figures 3 and 4 are examples of the type of data available from the COE. The data typically is in state-plane coordinates. To be useful for the survey, the coordinate units must be changed from feet to kilometers, and referenced to the local origin used in the survey. (The local origin is discussed in the following paragraphs.)

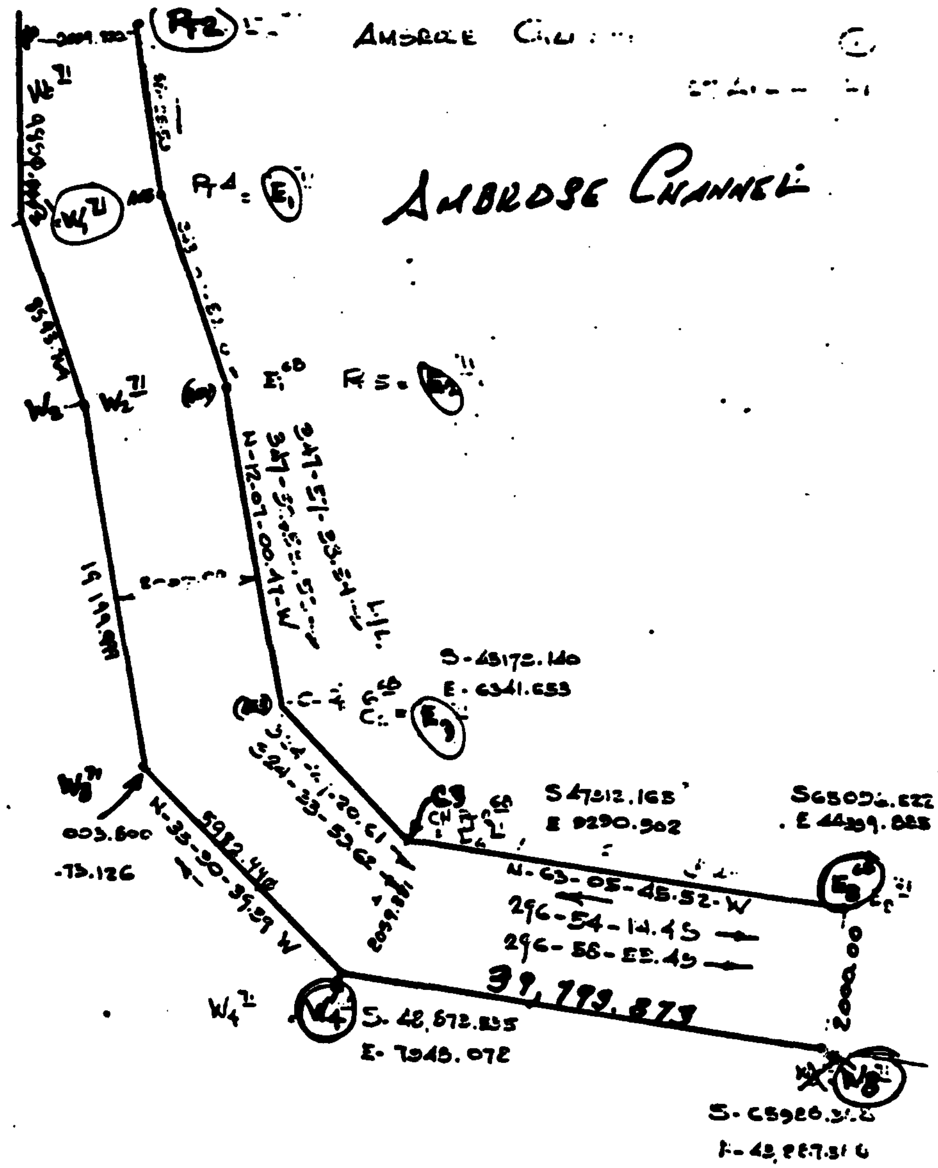
FEHG Algorithm

The Flat Earth Hyberbolic Grid (FEHG) TD to xy position conversion algorithm, developed for the PILOT Loran-C navigator[1], is used to validate waypoints. The algorithm uses a planar grid centered at an arbitrary local origin in the harbor area. The origin is usually chosen as a waypoint or a fixed aid to navigation near the center of the harbor area. The planar (xy) coordinates of transmitter locations are determined using great circle calculations based on the latitude and longitude of the transmitters and the latitude and longitude of the local origin. Waypoint xy coordinates (other than the local origin) are calculated from the difference in TD coordinates of the waypoint and local origin using the FEHG algorithm. (See APPROACH - Channel Edge Survey)

Waypoint and Loran-C Chain Data Files

The data analysis program for visual survey, TLS1, is used both during the presurvey planning and post-survey data analysis. Documentation and user instructions for TLS1 are contained in Appendix A. The program stores and uses Loran-C chain data and waypoint files to perform calculations. The Loran-C chain file contains the transmitter geodetic positions, transmitter power level, and secondary emission delays. The waypoint files store waypoint TDs, latitude, longitude, and xy position relative to the local origin. The waypoint file is a 25X8 array. Each row contains the data for one waypoint. Waypoint 25 is assigned to the local origin. Waypoints are broken down into convenient sequences and numbered. A waypoint file is created for each sequence. The number of waypoints in each sequence is usually limited to twenty. This allows four rows in the file for trackpoints to be stored if needed. (The local origin is stored in the waypoint 25 position in each waypoint file.)

During pre-survey the latitude and longitude of each waypoint is determined from the navigation chart and stored in the waypoint file. (Latitudes and longitudes can be calculated from state-plane coordinates using the algorithms listed in Reference 3.) The TDs for each waypoint are predicted using TLS1 and stored in the appropriate waypoint file. The xy coordinates of the waypoints based on the predicted TDs may also be calculated and stored in the waypoint file. This step may be used as a check of the position and predicted TD data previously entered.



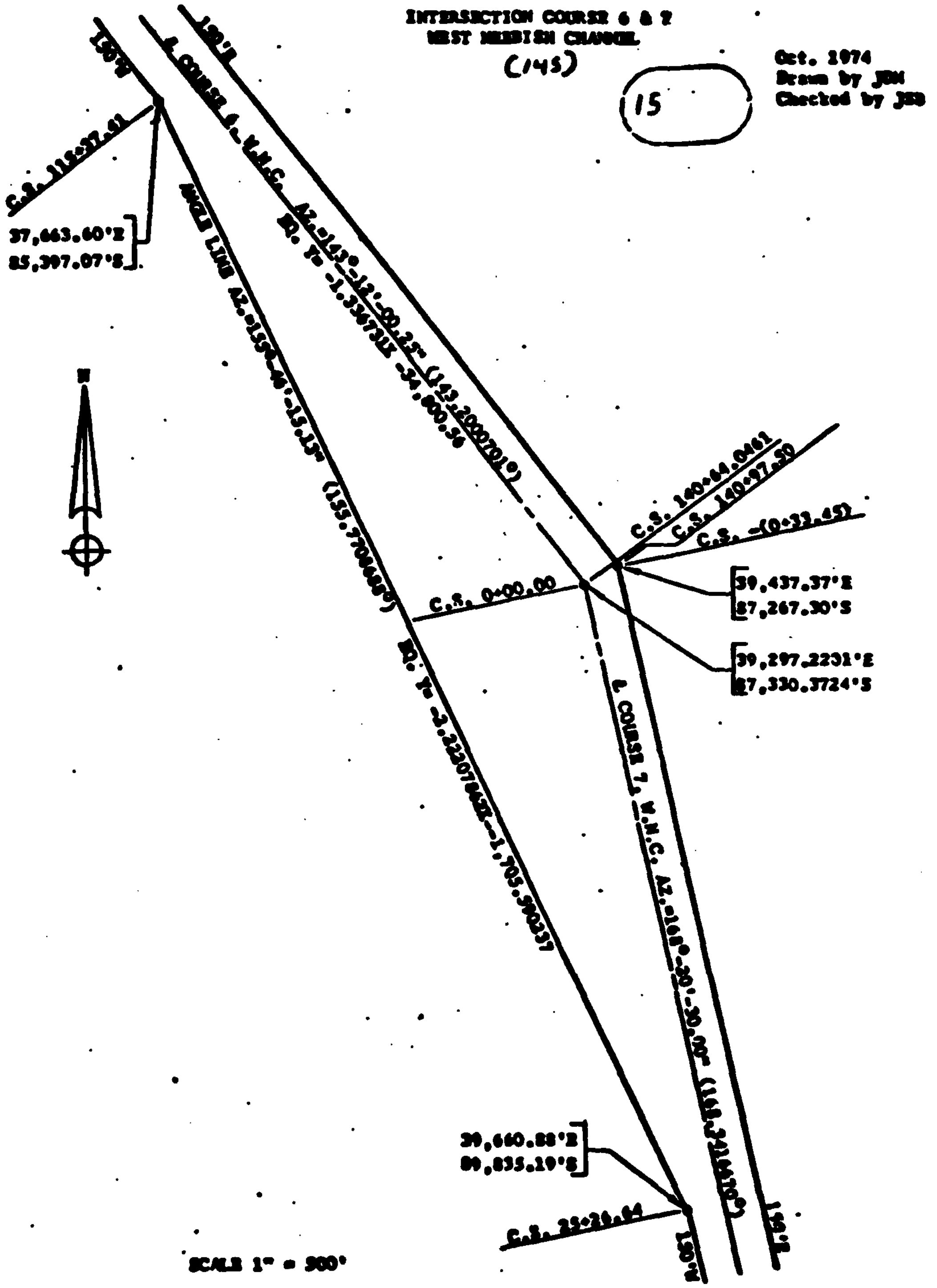
COE DREDGING DATA (NY HARBOR)

FIGURE 3A

TRAVERSE COMPUTATIONS												SHEET 3 OF 3 SHEETS
AMBROSE CHANNEL												
[illegible]				28 APRIL 1971								
	[illegible]	DISTANCE	[illegible]	[illegible]	LATITUDES		DEPARTURES		[illegible]	COORDINATES L.I.L.		[illegible]
					[illegible]	[illegible]	[illegible]	[illegible]		North	East	
									H C3	110,187.379	1997251.778	✓
[illegible]	40-46-36.55	[illegible]	.75731101	.65307731		1597.551		1366.95	W4	109,627.826	1995906.813	[illegible]
									R2	190,364.002	1985646.[illegible]	✓
[illegible]	81 58-75.5[illegible]	2069.8[illegible]	.14330170	.98937870		292.197		1985.46	W0	150,061.805	1962,857.71	[illegible]
									E6	18,355.[illegible]	6,098,874.[illegible]	✓
[illegible]	26-26-[illegible]	[illegible]	.89119287	.45590317		1782.50[illegible]		907.406	W0	10,573.[illegible]	[illegible]	✓
[illegible]	116-50-23.40	18793.873	.45370516	.89115267	18[illegible]			9542.[illegible]	W4	108,627.[illegible]	1,995,906.8[illegible]	✓
[illegible]	144-33-51.62	5907.440	.81478961	.57972678	4876.430			3418.96[illegible]	W3	113,502.[illegible]	1993,438.48[illegible]	✓
[illegible]	167-57-25.5[illegible]	[illegible]	.97800177	.20652[illegible]	18777.6[illegible]			4004.[illegible]	W2	132,279.[illegible]	1,989,4[illegible]	✓
[illegible]	163-07-[illegible]	[illegible]	.95718105	.28[illegible]	8215.535			[illegible]	W1	140,505.[illegible]	1,985,[illegible]	✓
[illegible]	165-58-[illegible]	9650.000	.97015489	.24236599	9556.[illegible]			2397.[illegible]	W0	150,061.805	1983,557.[illegible]	✓

COE DREDGING DATA (NY HARBOR)

FIGURE 3B



COE DREDGING DATA (ST. MARYS RIVER)

FIGURE 4

Preparation of Survey Plan

A survey plan is prepared for each waypoint using the navigation charts, COE survey data, and information gathered from local A-to-N personnel and users of the harbor area.

Range-Range Waypoints.

Figure 5 is a survey planning form for a waypoint defined by the intersection of two ranges. Items to consider for a range-range survey are:

a. survey pattern: an "X" survey pattern is the preferred survey pattern. Unfortunately this pattern is not always realizable due to the lack of navigable water or the loss of visibility of one of the range markers beyond the waypoint. It may be possible to run a balanced survey line (e.g., bracket the waypoint) on one trackline, and not on the other. This pattern, e.g., a "Y", is better than a V. An effort should be made to collect data at least to the waypoint. In some cases even this may be difficult as the front or rear range may become obscured near the waypoint.

b. trackline length. The recommended survey trackline length is approximately 1 kilometer. The anticipated tracklines endpoints should be marked on the navigation chart (or a copy of the chart in the area of the waypoint) and turn aids noted.

c. potential problems. The navigation chart should be studied carefully for potential problem areas such as bridges and overhead power lines near the tracklines. Avoid the region for at least 300 meters to either side of a power line or bridge when collecting waypoint data. Range markers that will be at the far end of the channel should be noted. These ranges may be difficult to see during daylight hours. Such waypoints are much easier to survey during dusk or at night.

d. simulation. One of the program functions of TLS1 is a range-range survey simulation. Input are the waypoint number, start and stop points of the survey tracklines, expected TD noise, and number of samples. The program function computes a data set for each trackline and the resultant waypoint calculations. The simulation is useful for predicting the crossing angles of the tracklines, linear regression line slopes, and the effect of changing survey patterns. In general, shallow crossing angles of tracklines on the navigation chart will also produce shallow crossing angles in TD coordinates. The only method to compensate for this is to increase the number of data samples. The predicted regression line slopes are helpful during field data collection as a check of the reasonableness of the data as it is being collected (see Data Collection).

Channel Edge Survey.

The navigation chart for the area near the waypoint must be studied to determine the aids to navigation channel edges and where TD data will be collected. In general the more data collected, the better; however, some effort must be made to keep the amount of data and time spent collecting the data within reason. Some planning is advisable for the method the data will be stored in data files. It takes a little longer to store data into several files rather than one larger file during the data collection phase, but it is easier to combine data files than break them up during data analysis. A recommended procedure is to store data collected at each aid to navigation and along channel edges (e.g., one side between two waypoints) in separate files.

SURVEY PLANNING FORM

VISUAL RANGE SURVEY

#WP ID 24 WP FILE SHRI WP # 13

WP DESCRIPTION INTERSECTING RANGE — LOWER NICOLET RANGE &

Y-TYPE UPPER NICOLET RANGE

EST LAT/LON LAT 46° 23' 59.76" N LON 84° 14' 0.42" W

SURVEY STRATEGY: TRACKLINE LENGTH 1½ KM

TURN POINTS : TRACKLINE ONE BUOY 76 AND WP & EQUAL TIME

TRACKLINE TWO BUOY 82 AND WP & EQUAL TIME

SIMULATED SURVEY COURSE:	WX	WY	WZ	XY	XZ	YZ
SLOPE ( )						
SLOPE ( )						
CROSSING ANGLE						

	W	X	Y	Z
PREDICTED TD'S		30971.258	47893.006	59134.921
EST ASF		-.071	-.253	+1.733
PRED TD'S CORR FOR ASF		30971.187	47892.753	59136.654
SURVEYED TD's				

DATA FILES: WAYPOINT SURVEY

TAPE #	FILE	DATE	COMMENTS

DATA FILE: VERIFICATION

TAPE #	FILE	DATE	COMMENTS

PLANNING FORM FOR RANGE-RANGE SURVEY

FIGURE 5

Verification Data.

Verification data for range-range waypoints can be readily collected when traveling from one waypoint to another and enroute to a survey area. If the same area is traveled several times, data can be taken along the channel edges as well as along the channel centerline.

It is a good practice to plan to collect a redundant set of channel edge or fixed aid-to-navigation data in the area of a waypoint defined by channel edge data. This can be used as a check for the waypoint calculations.

DATA COLLECTION

Time Difference Survey System (TDSS)

The TDSS is a data acquisition system designed to collect Loran-C time difference and position data for Loran-C harbor survey applications. The system provides a real time graphic display of the data as it is collected and calculates cumulative statistics including linear regression parameters for the TD variables.

The TDSS consists of a Hewlett-Packard HP9845T desktop computer, an Austron 5000 Loran-C monitor receiver, a Motorola Mini-Ranger positioning system and a Deltec Uninterruptable Power Supply (UPS). The Mini-Ranger system is not used for visual survey applications. The UPS is necessary only when the reliability and stability of electrical power is in question.

A complete description of the TDSS is contained in the Time Difference Survey System (TDSS) Installation and Operation Manual[2].

Special Function Keys.

Figure 6 is the overlay for the special function keys (SFKs) on the HP9845. The keys can be functionally divided into two groups: Austron 5000 Control and Data Collection. Tables 1 and 2 are brief descriptions of the functions of the SFKs for each group.

TD/TD Graphics Plot.

An example of the TD/TD graphics plot is shown in figure 7. The TDs to be plotted, predicted waypoint, anticipated minimum and maximum TD values, and predicted regression line slope must be input to the calculator during data collection initialization. Each data point collected is plotted on the CRT (if it falls within the plotting area). The cursor is positioned over the latest sample. The bar graphs to the left of the plot indicate the confidence bound of a regression line fit to three pairs of TDs (WX, XY and YZ). Note that this is the confidence bound at the mean (approximately the center) of the trackline. The confidence bound at the end points is approximately twice this value.

Statistics Tabulation.

At the end of a data collection run (typically 100 samples with a maximum of 400 samples), the graphics display is dumped and a statistics summary table is printed on the hard copy printer. Figure 8 is an example of the statistics summary. A printout of the current statistics can be obtained at anytime during a data collection run by pressing the Stats key (SF28).

Austron 5000 Setup.

The Austron 5000 Loran-C Receiver offers operational flexibility not found in a receiver designed for navigational use. Some important features for survey use are:

SPECIAL FUNCTIONS — MASS STORAGE IS

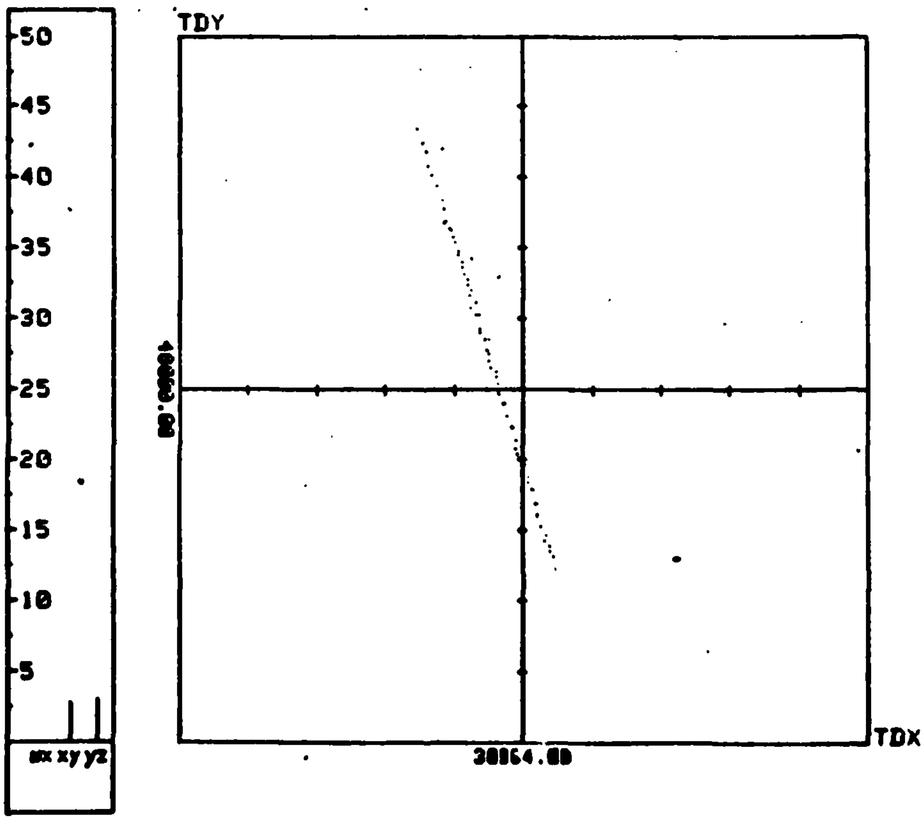
S	GRAPHICS	DUMP GRAPHICS	ACQ1	RPRT1	TMCN	Data-Coll	:T14	:T15
	K0	K1	K2	K3	K4	K5	K6	K7
	PRINTER 0	PRINTER 16			OFF Kbd		REWIND T14	REWIND T15
S		STRT	STOP	DEBUG	STATS	SIGN1		
	K8	K9	K10	K11	K12	K13	K14	K15
	On-Kbd	PAUSE.	CONT			EDIT LINE	LIST	

NOTE:
S = SHIFT KEY HELD PRIOR TO PRESSING S FUNCTION KEY, WHICH IS KEYS K16→K31

SPECIAL FUNCTION KEY OVERLAY

TDSS SPECIAL FUNCTION KEY OVERLAY

FIGURE 6



EXAMPLE OF TDSS GRAPHICS PLOT

FIGURE 7

a. TMCN: tracking loop bandwidth (time constant) - the TMCN command controls the tracking loop time constant (TMCN/10 seconds). The higher the TMCN (longer time constant) the greater the noise rejection, and the poorer the tracking performance during changes in course and speed. For survey applications it is desirable to have low TD standard deviations (high noise rejection) and good tracking performance. Another constraint for data collection is that data samples should be independent (e.g., not time correlated). This latter constraint requires the interval between data samples to be at least two tracking loop time constants (e.g., 2XTMCN/10 seconds). The maximum sampling rate of the TDSS is one sample every 12 seconds. This limitation is caused by the data transfer rate of RPRT messages from the Austron 5000 to the HP9845. A TMCN of 50 (approx. 5 seconds) provides good signal-to-noise ratio performance (standard deviation = 20-50 nanoseconds), acceptable dynamic performance, and enables the TDSS to be operated at its fastest sampling rate.

b. CLIP: linear clipping level - noise in the low frequency spectrum tends to be burst-like rather than continuously Gaussian distributed. The Austron 5000 receiver provides linear processing which has excellent performance characteristics under many conditions, (and is optimum for Gaussian noise) but is suboptimal for burst type noise. There is a provision for non-linear processing in the Austron 5000 which considerably improves receiver performance in the presence of burst-type noise. The process is clipping and is set by the CLIP command. A CLIP level of 130 is recommended for survey operations.

c. STATIONS: stations tracked - The Austron 5000 has the capability of tracking several Loran-C chains simultaneously and up to a total of six stations. The TDSS software is designed for the Austron to be operated with one chain with up to five stations including the Master. It is not necessary that all four secondaries be in the track mode.

Waypoint Data, Range-Range

Track-keeping.

The ranges which define the waypoint and planned survey tracklines are determined during the Presurvey Planning. On arrival in the area of the waypoint, the ranges and turn aids should be identified. As discussed in Presurvey Planning, the recommended trackline length is approximately 1 kilometer (1/2 nautical mile). It is impossible to keep the ranges perfectly aligned during an entire data collection run. However, an attempt should be made to keep the survey vessel on range as close as possible. Wind and current will tend to bias the vessel to one side of the centerline. The tendency is to fall off the range, correct to the centerline, fall off the range, etc. This process tends to bias the data to the down-wind (current) side of the channel. A preferred procedure is to correct the vessel track slightly to the up-wind (current) side of the channel centerline; such that the average trackline is on the range. It is not necessary to "pause" data collection when the survey vessel is slightly to one side or other of the channel. Data collection should be "paused" during turns and when the survey vessel is significantly off the range.

Sample Number and Multiple Runs.

The recommended survey procedure is to collect at least two repetitions of the survey trackline rather than attempting to collect the desired number of

COMMAND	KEY	FUNCTION
ACQ1	K18	-Transmits an ACQ1 command (signal acquisition) to the Austron 5000
RPRT1	K19	-Transmits a RPRT1 command to the Austron 5000
TMCN	K20	-Requests a TMCN value from operator, then sends command to Austron 5000 to set input TMCN to all stations
ON-KBD	K8	-Sets HP9845 keyboard in teletype mode. Keyboard is used as a terminal to control Austron 5000 using standard Austron commands.
OFF-KBD	K4	-Returns HP9845 keyboard to normal calculator mode.

TDSS SPECIAL FUNCTION KEYS RELATED TO AUSTRON 5000
LORAN-C RECEIVER CONTROL

TABLE 1

COMMAND	KEY	FUNCTION
Data-col	K21	-Initializes data collection parameters.
Start	K25	-Starts data collection.
Pause	K9	-Pauses data storage and plot; samples are displayed on CRT.
Cont	K10	-Cancels data-collection Pause.
Stop	K26	-Stops data collection. The graphics display is dumped onto the thermal printer and a statistics summary table is printed. Data may be stored on magnetic tape.
Stats	K28	-Prints an interim statistics summary table.
Sign	K29	-Inverts the trilateration triangle for calculating position from Mini-Ranger data; used when baseline between reference stations is crossed.

TDSS SPECIAL FUNCTION KEYS RELATED TO DATA COLLECTION

TABLE 2

samples on a single run. However, if conditions permit accurate trackkeeping at a slow speed, a single run is adequate. The desired number of samples depends on:

a. the distance to the front range and sensitivity of the range. This will determine the capability of keeping the vessel on the channel centerline (track-keeping noise).

b. the crossing angle of the tracklines. The only method to compensate for poor crossing angles is to increase the number of data samples.

c. the survey pattern. If a "V" pattern must be used, a larger number of samples is required than if an "X" pattern is used.

The usual range of samples is 50 to 200. However, there is usually not much to be gained beyond 100 samples. The bar graphs on the TD/TD graphics display are a good indicator of data quality. If the bar graph for the TD pair of greatest interest is between 5 and 10 nanoseconds, there is usually sufficient data. Another good check of data quality is the correlation coefficients for the data pairs. Correlation coefficients in the range of $\pm.9$ to $\pm.999$ can be expected.

Care should be taken in collecting data with the range markers "over the shoulder." It is sometimes advantageous from a survey standpoint to collect data in both directions. However, it is considerably more difficult to maintain a good vessel track with the ranges to the stern of the survey vessel. In some cases it is more prudent to collect data only in the approaching direction. It may add some time to the data collection phase, but it greatly simplifies the data analysis if little or no data editing is necessary.

Verification Data, Range-Range

Data to verify waypoint calculation is collected along the channel centerlines over the entire length of the channel. It is important during the data analysis phase to know when the vessel was on the centerline. Notes should be made during the survey of the sample numbers and approximate alongtrack distance when the survey vessel is on the range. It will not always be physically possible to maintain the vessel on the range over the entire length of the centerline due to other vessel traffic. Again, the sample number of when the vessel deviated from the centerline and when it returned are extremely useful during data analysis. If the survey vessel must deviate from the centerline for other traffic, it is a good practice to move to the channel edge until the centerline is clear. The distance that the vessel moved calculated from the TD data is compared to distance between the channel centerline and channel edge on the navigation chart or COE data during data analysis.

The data for waypoint verification can be collected enroute between waypoints and enroute to a survey area. Several validation runs provide a good check on the waypoint and the variability of the TD grid. One or more centerline data runs and data taken along both channel edges are usually sufficient.

If the verification data is being collected while enroute to a survey area, the data for several channels can be strung together in the same data file. The sample number when a turn is started and when the vessel is steady on the next range should be recorded to assist the data analyst. The TDSS can collect up to 400 samples in a single data file. If this number will be exceeded with the vessel partially between two waypoints, the data collection run should be stopped, the data stored, and a new data collection run started.

Waypoint Data, Channel-Edge

Data to calculate the waypoint is collected at principal aids-to-navigation near the waypoint and along the channel edges of the two channels which define the waypoint. Wind, current and the position of the survey vessel with respect to the aid-to-navigation should be recorded when collecting data at an aid-to-navigation. The average fathometer reading should also be recorded.

Data collected at aids to navigation near the waypoint should be stored on separate data files. The channel edge data should be collected and stored in segments consisting of one side of the channel between two waypoints. Buoys or fixed aids along the channel edge can be marked by stationing the survey vessel near the aid until 5 to 10 samples are collected. All accessible fixed aids and every second or third buoy should be marked.

Waypoint Verification, Channel-Edge

Waypoint verification for the channel edge survey is accomplished as part of the waypoint calculation. A redundant data file for at least one of the channel edges should be collected. This data set can be used during data analysis to check the waypoint calculations.

Data Files

The TDSS stores data in files on a magnetic tape cartridge with a maximum of 400 samples per file. Each file is identified with a file name consisting of 1-6 alphanumeric characters. It is convenient to establish a convention for naming files before a survey begins so that the data that is stored on the file can be identified for data analysis. Some suggestions for naming files are listed below:

a. Range-Range Survey Line (Tracklines): TL (waypoint from) (waypoint surveyed). TL76 and TL56 would be the two survey files for waypoint 6.

b. Range-Range Verification: CL(Start Waypoint) (Stop Waypoint). CL1819 is the data collected on the centerline (CL) between waypoints 18 and 19, CL1720 is the data collected on channel centerlines starting near WP17 and stopping near WP20.

c. Channel Edge Data: RE(Start waypoint) (Stop Waypoint). RE1314 is the data along the channel edge marked by red buoys (RE) between waypoints 13 and 14; BE1314 is the data along channel edge marked by black buoys (BE) between waypoints 13 and 14. BUOY15 is data collected at Buoy 15, LT28 is data collected at Light 28.

d. Dockside Data: DS(Julian Date)(Number). DS1892 is the second set of dockside data collected on Julian day 189.

Local Monitor

Data from a local monitor is collected so that a correction plot can be prepared for the time interval when survey data was collected. An initial comparison can be made of dockside data and the TD data from the local monitor during the periods when dockside data was collected before getting underway and after returning to port. This comparison is a check that any offsets in the dockside data were also observed at the local monitor. This provides a check of both the TDSS and local monitor. The correction plot is used during data analysis to remove any temporal offsets from the survey data.

DATA ANALYSIS

A Hewlett Packard HP9845T desktop computer and the computer program "TLS1" are used to analyze the data collected during a harbor survey using visual reference positioning techniques. The program "TLS1" is a collection of special function programs which can be used to

a. calculate waypoint TDs, and

b. evaluate the performance of the resultant waypoints.

TLS1. The program TLS1 consists of 19 special function programs selected by one of the special function keys of the HP9845. The functions are divided into six groups and a short description of each program function is provided in the following sections.

Presurvey Planning Group

a. Predict TDs, K20: calculates predicted TDs for an input latitude, longitude. The program also outputs range and bearing to transmitter stations and GDOPs for three TD and two TD position fixes. It requires that a Chain Data File be available.

b. File or Read Waypoint Data, K23: used to create waypoint table files and to edit and restore waypoint table files. It can also be used to obtain a listing of the waypoint table. The waypoint table is a 25X8 matrix. The first four columns contain waypoint TD values (TDW, TDX, TDY, TDZ). Columns five and six contain the xy position referenced to the local origin. Columns seven and eight contain the latitude and longitude in decimal degrees. Waypoint 25 (e.g., row 25 is designated as the local origin. The latitude and longitude stored in this location is used by program functions which calculate the local xy coordinates of the transmitters. If the local origin is also one of a sequence of waypoints, its parameters will be stored twice, i.e.; in row 25 and in the row corresponding to its logical waypoint number.

c. Store Chain Data, K6: stores Loran-C chain data (transmitter geodetic positions, transmitter power levels, and secondary emission delays) on a data file for later use. File names are a five character mnemonic for the particular Loran-C chain. The first four characters are an abbreviation for the chain (e.g., NEUS, SEUS, GTLK, etc.). The fifth character is a number from 1 to 4 which designates the configuration of three TDs: 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ.

d. Simulate Waypoint Survey, K28: simulates the survey of a waypoint using the intersection of visual ranges technique. Two TD data sets are generated along the tracklines defining a waypoint. Input variables are the waypoint of interest, the starting and stopping points of the survey lines, and the expected standard deviation of the TDs. The program function processes the data as if it were field data. Statistics and regression parameter tables are printed on the hard copy printer for each data set along with the waypoint table. The "true" TD values for the waypoint are zero. The tabulated waypoint value is the error in estimating the waypoint.

Survey Data Handling Group

a. Read Data File, K0: reads time-difference (TD) and time-of-day data stored on magnetic tape. The TD data may be corrected for known offsets. If TDs are corrected, a lower case "t" is automatically annotated to the file name.

b. Edit Data, K16: used to edit (i.e., remove) samples from the data (TD, time, and range) arrays. Three options are available to the user:

(1) The first option deletes a single data sample in each data array.
(2) The second deletes a block of data samples.
(3) The third deletes samples with TD samples outside a range that is input by the operator.

When editing of the data arrays is complete, the operator may store the edited data on a new file for later use.

NOTE: The edit function deletes only TD data samples. Arrays previously formed containing XY and along/cross track position data are not affected. Functions K1 and K17 must be repeated to see the effect of editing on computed positions.

c. Separate Data into Subfiles, K3: allows a large data file to be broken up and stored in several smaller data files or a subset of a large data file to be stored as a separate file.

d. Link Data Files, K25: enables multiple data files to be loaded into memory. TD data can be corrected for each file entered. The total number of samples must be equal to or less than 400. The function will automatically ignore any samples which would cause this limit to be exceeded.

e. Read Data from MFE5000, K29: reads TDs collected from an MFE 5000 tape drive. The data is assumed to have been recorded from an Internav 404 EIA OUT port with the following data format:

Characters 1-8; GRI
Characters 10-17; TDA
Characters 19-26; TDB
Characters 28-35; TDC
Characters 37-44; TDD

The program function also assumes that TDW = TDA; TDX = TDB, TDY = TDC; and TDZ = TDD. If this is not true, program lines 1820 through 1850 must be changed to reflect the proper arrangement. This program is normally not used with the TDSS. It can be used to analyze data collected using a supplementary LC404 receiver and a MFE 5000 data recorder.

Statistics and Regression Group

a. Calculate Statistics and Linear Regression of TD Data, K2: calculates statistics, linear regression coefficients, and minimum and maximum for the four TD arrays. Statistics and regression parameters calculated are:

(1) mean for each TD
(2) standard deviation for each TD
(3) covariance of TD pairs
(4) correlation coefficient of each TD pair
(5) linear regression slopes of each TD pair
(6) standard deviation of residuals for each regression line

b. Plot TD Data with Regression Lines, K4: plots two of the time-difference arrays against each other and also plots the linear regression line for the TD pair chosen. The program automatically scales the plot for the minimum and maximum TDs for each array. The axes are drawn through the mean for each TD. Minor tic marks are every microsecond and major tic marks every 10 microseconds.

c. Plot Residuals, K5: plots the residuals from the linear regression for any TD pair (See K2). The residuals may be plotted against sample number or the independent variable. Residuals are normalized to the standard deviation of the residuals. Normalized values greater than 5 are printed on the hard copy printer and are not plotted.

Waypoint Calculation Group

a. Calculate Waypoint, K21: computes range-range waypoint time differences, estimated rms error, and lop crossing angles. The function also calculates and prints statistics and regression parameters for each of the survey tracklines.

b. Daisy Chain, K22: calculates the position of a waypoint based on the difference in TDs between it and a neighboring waypoint. The Flat Earth Hyperbolic Grid (FEGH) is used to calculate position coordinates. Differential xy coordinates, differential latitude and longitude, and range and bearing between waypoints are also calculated. The user has the option of inserting the calculated xy coordinates (and latitude longitude) into the waypoint table, Wpt(*). Note: This function does not restore the new waypoint table in magnetic tape. If this is desired, function K23 must be used.

c. File or Read Waypoint Data, K23: used to create waypoint table files and to edit and restore waypoint table files. It can also be used to obtain a listing of the waypoint table. The waypoint table is a 25X8 matrix. The first four columns contain waypoint TD values (TDW, TDX, TDY, TDZ). Columns five and six contain the xy position referenced to the local origin. Columns seven and eight contain the latitude and longitude in decimal degrees. Waypoint 25 (e.g. row 25) is designated as the local origin. The latitude and longitude stored in this location is used by any program function which calculates the local xy coordinates of the transmitters. If the local origin is also one of a sequence of waypoints, its parameters will be stored twice, i.e. in row 25 and in the row corresponding to its logical waypoint number.

d. TD Move, K24: calculates the change in TD from a waypoint to a position offset from the waypoint. This offset in waypoint position and TD may be applied to the waypoint table. The change in TDs is calculated based on the change in distances to the transmitters.

Data Conversion Group

a. Convert TD Data to XY and Along/Cross Track Position, K17: converts TD data to xy and along/cross track positions using the FEHG algorithm and surveyed waypoints. The program will compute either a three or two TD solution. A summary table is printed on the hard copy printer which lists:

-chain and LOPs used in the solution
-file name
-bearing angle between waypoints used for along/cross track calculation
- rms trackline of data
-average cross track position
-standard deviation of cross track position
-average xy position
-standard deviation of xy position data

b. Plot XY Data, K18: This program function plots the xy data calculated from Loran-C TDs on the CRT or 9872A plotter. Two options are available. The first automatically scales the CRT (or9872A) plotting area to the range of xy data. The operator may zoom in on a section of the plot, find the sample number of a plotted point, and digitize up to 10 locations on the plot. The second option plots the xy data to a chart scale (1:10,000,1:20,000, 1:40,000, or 1:80,000). The axes are drawn through a waypoint selected by the operator. The operator also inputs the offset of the waypoint (axes) from the lower left hand corner of the plotting area. The selection of waypoint offset and scale determines the window of data which will be plotted.

c. Plot Along/Cross Track Data, K19: This program function plots cross track vs along track position. The plot is automatically scaled and labeled.

XY to TD Offset Conversion Group

a. TD Move, K24: See Waypoint Calculations

b. Measured-Projected TDs, K27: calculates the statistics of the difference between the TDs measured and TDs projected from the waypoint based on the xy position calculated from the measured TDs. The result is a measure of the three TD fix triangle. The function can be used to estimate the third TD of a waypoint if two are known.

<u>Waypoint Calculations</u>

Range-Range.

For the range-range survey case, data is collected on each range line and stored in data files on a magnetic tape cartridge. The term trackline is used for the survey line along one of the visual ranges. Trackline data is collected near the waypoint and optimally brackets the waypoint. A trackline data file typically contains two to four runs along the range line. For each pair of TDs collected, a straight line is fit to the trackline data using linear regression. The waypoint is calculated as the intersection of the regression lines for each trackline.

The function Compute Waypoint, (K21), calculates statistics and linear regression parameters for each TD pair of data collected for the two trackline data files and the intersection of the regression lines for each TD pair. The crossing angle of the regression lines and the estimated rms error of the waypoint estimate (intersection of regression lines) are also calculated. The results are printed in tabular form as shown in figure 9.

The quality of the trackline data can be checked by examining the correlation coefficient and residuals for each TD pair. Four TD pairs can be collected, but it is rare that all four provide useful data. In many cases, there are three good TDs available. The correlation coefficient and residuals are a measure of how good the trackline data fits a straight line. If the vessel track were a perfect straight line, the residuals would be slightly greater than the standard deviations of the dependent variable observed at dockside. A "noisy" vessel track will show up as larger residuals. A typical range in residuals is .025 to .060 microseconds. One's ability to detect if the survey vessel is on or off the range decreases as the distance to the range markers increases. Therefore, tracklines run at the far end of the range will tend to have larger residuals than tracklines nearer the range markers. In some cases, portions of tracklines run with the range markers "over the shoulder" may cause problems due to poor track keeping. The survey officer's notebooks should be checked for comments concerning suspect data. Suspect data can be removed using the Edit (K16) function. TD Statistics and Regression (K2), Plot TD Data with Regression Lines (K4) and Plot Residuals (K5) are useful functions for detecting suspect data and identifying sample numbers to delete.

The estimated rms error and the closure between waypoint solutions is a good indication of waypoint quality. Estimated rms error is a function of trackline residuals, crossing angles, number of samples, and the survey pattern. Reducing trackline residuals by editing suspect data will also reduce rms error in the waypoint solution. (Note, the number of samples is also reduced.) Crossing angles are fixed by the configuration of the channel and the Loran-C chain geometry. The rms error is inversely proportional to the square root of the number of samples. The error is minimum when the survey tracklines form an X pattern. The difference between an X pattern and a V pattern with the same number of samples is a factor of two. Editing data to better bracket the waypoint can improve the calculated rms error.

If there are three TD data arrays, there are two solutions for the waypoint value of each TD. If four TDs are collected, there are three solutions for each waypoint TD. The term, closure, is used to describe the agreement between solutions. If two solutions have low estimated rms errors (e.g., less than 20 nanoseconds), there should be good closure between the solutions. A good rule of thumb is that the closure difference should be less than twice the sum of the estimated rms error for each solution.

	<TDW>	<TDX>	<TDY>	<TDZ>
CUMULATIVE AVERAGE	0.000	30963.545	40000.359	59196.777
STANDARD DEVIATION	0.000	.541	1.694	.053

	<WX>	<WY>	<WZ>
CORRELATION COEFFICENT	9.99999999999E+99	9.99999999999E+99	9.99999999999E+99
REGRESSION LINE SLOPE	9.99999999999E+89	9.99999999999E+99	9.99999999999E+99
RESIDUAL	0.000	0.000	0.000
INDEPENDENT VARIABLE	2	2	2

	<XY>	<XZ>	<YZ>
CORRELATION COEFFICENT	-.999	-.998	1.000
REGRESSION LINE SLOPE	-3.135	-1.500	.503
RESIDUAL	.023	.031	.025
INDEPENDENT VARIABLE	2	2	1

SAMPLES= 69

EXAMPLE OF TDSS STATISTICS SUMMARY

FIGURE 8

TRACKLINE ONE= TL1017

	TDW		TDX		TDY		TDZ
CUMULATIVE AVERAGE	0.000		27401.153		43356.355		59504.551
STANDARD DEVIATION	0.000		1.905		1.190		1.699

TD PAIR	WX	WY	WZ	XY	XZ	YZ
CORR COEF	0.000	0.000	0.000	-1.000	-1.000	1.000
SLOPE	0.000	0.000	0.000	-.603	-.056	1.419
RESIDUAL	0.000	0.000	0.000	.036	.050	.034
IND VAR	2	2	2	1	1	2

SAMPLES= 120

TRACKLINE TWO= TL1617

	TDW		TDX		TDY		TDZ
CUMULATIVE AVERAGE	0.000		27402.333		43355.075		59503.600
STANDARD DEVIATION	0.000		.065		.040		.410

TD PAIR	WX	WY	WZ	XY	XZ	YZ
CORR COEF	0.000	0.000	0.000	.791	-.994	-.750
SLOPE	0.000	0.000	0.000	.037	-.471	-13.515
RESIDUAL	0.000	0.000	0.000	.024	.044	.026
IND VAR	2	2	2	1	1	2

SAMPLES= 44

WAYPOINT SOLUTION

TD PAIR	WX	WY	WZ	XY	XZ	YZ
TDW	0.000	0.000	0.000			
TDX	0.000			27401.970	27401.975	
TDY		0.000		43355.062		43355.063
TDZ			0.000		59503.040	59503.053
RMS ERROR	0.000	0.000	0.000	.010	.033	.009
CROSSING ANGLE	0.000	0.000	0.000	33.203	15.310	140.507

EXAMPLE OF RANGE-RANGE WAYPOINT CALCULATIONS

FIGURE 9

The waypoint TD estimate with the lowest estimated rms error is chosen as the waypoint value. If two solutions have approximately the same rms error, they are averaged.

Figure 10 is a summary of the data analysis procedure for calculating a waypoint with range-range data.

If one of the three waypoint TDs calculated using the range-range approach has poor rms estimates for all solutions, a supplemental technique can be used. Here the xy position coordinates for one of the survey tracklines or an aid-to-navigation near the waypoint can be calculated using Convert TD Data to XY and AT/CT (K17), the estimated waypoint, and the two TD solution (using the two good TDs). The function, Measured-Projected TDs (K27), calculates a set of three TDs for each xy data point based on the difference in distance between the waypoint and xy data point and the TDs of the waypoint. These "projected" TDs can be subtracted from the measured TDs for each sample and the mean and standard deviation of this difference calculated. The difference data for the two TDs used for the xy solution will be near zero (the "project" function is the inverse of the FEHG algorithm). The difference data for the third TD is the correction that must be applied to the waypoint value which will force it to be consistent with the other two TDs. This correction can be applied and the waypoint TDs are restored (K23). The calculations can then be checked by using the three TD solution (K17) to convert the other trackline data file to xy positions. The function, Measured-Projected TDs (K27) is applied to this data set. The result is a measure of the average fix triangle for the data set. Figure 11 is a summary of this supplemental data analysis procedure.

Channel Edge

An iterative approach must be used for the channel edge survey case. TD data should be collected along channel edges, the channel centerline (if a range is available), and at aids-to-navigation near the waypoint and along channel edges. The TDs of a fixed aid near the waypoint (a buoy may be used if it is the only aid available) should be stored temporarily as a first estimate of the waypoint TDs (K23). The corresponding xy coordinates are calculated using the Daisy Chain (K22) function from the local origin. The offset of the waypoint from the aid-to-navigation can be estimated from the navigation chart for the area and used in the Move (K24) function to calculate the estimated waypoint TDs.

The TD data collected in the area around the waypoint can be converted to xy and along/cross track position coordinates using function K17, Convert TD Data to xy and AT/CT. The converted data can be plotted and compared to the navigation chart. Function K18, Plot XY Data, will plot the position data on the CRT or 9872A plotter. The plot can be scaled to a navigation chart scale (e.g., 1:10,000, 1:20,000, 1:40,000, 1:80,000) or automatically scaled for the range of xy data. The function K19, Plot Along/Cross Track Data, automatically scales and plots the calculated cross track vs along track data.

- Calculate intersection of survey tracklines (K21)

- Check quality of trackline data and waypoint solution
  - correlation coefficients (.9 - .999)
  - trackline residuals (.020 - .060 microsec)
  - estimated rms error or waypoint solutions (function of residuals, sample number, crossing angle, and survey pattern)
  - closure of waypoint solutions (difference between waypoints estimates less than twice the sum of estimated rms errors)
  - crossing angles

- If data is not satisfactory, edit suspect data from data files and restore (K16), and recompute waypoint (K21).

- If data still not satisfactory, resurvey waypoint.

- Choose the waypoint estimate with the lowest rms error as the waypoint value (estimates with approximately the same estimated rms error may be averaged) and store in the waypoint table (K23).

- Calculate xy position of waypoint referenced to local origin and store in the waypoint table (K24).

- Restore the waypoint table in the waypoint file (K23).

Figure 10
Waypoint Calculation, Range-Range Data

- Calculate and store waypoints TDs using standard range-range approach (figure 10)

- Convert the TDs for one of the survey tracklines or a nearby aid-to-navigation to xy coordinates (K17) using the two TD solution and two good TD estimates.

- Use Measured-Projected TD function (K27) to calculate a correction for the third TD

- Apply correction and restore waypoint TDs in waypoint table (K23)

- Convert TDs for other survey line to xy coordinates (K17) using the three TD solution

- Use Measure-Projected TD function (K27) to calculate average fix triangle

- If results not reasonable, recheck calculations

- Restore waypoint table on waypoint file (K23)

Figure 11
Supplemental Range-Range Waypoint Calculation

The xy and along/cross track plots can be compared to the navigation chart (and COE dredging charts if available). If the plotted data is consistent with the chart, then the calculated waypoint TD coordinates are good. If the plotted data and navigation chart do not agree, the offset of the waypoint can be determined from the plots. The waypoint can then be "moved" using K24 and the xy and cross track data can be recalculated and replotted. This process must be repeated until the plotted data agrees with the navigation chart and COE data.

Figure 12 is a summary of the data analysis procedure for calculating the TDs for a waypoint with channel edge data.

- Calculate the average TDs for a data file collected at an aid-to-navigation near the waypoint (K2).
    - If the survey officer's notebook indicates that some data samples are suspect, edit these samples (K16) before calculating statistics.
    - If the standard deviation of the TDs are significantly higher than observed at dockside, check the data for outliers using the Plot TD (K4) and Plot Residuals (K5) functions and edit suspect data (K16).
- Store the TDs for the aid-to-navigation near the waypoint as a temporary waypoint (K23).
- Calculate the xy coordinates of the temporary waypoint using the local origin as the reference point (K22).
- Offset the TDs and position from the temporary waypoint to the waypoint using a differential position estimated from the navigation chart or COE data (K24).
- Convert TD data collected along channel edges, channel centerlines, and at aids-to-navigation to xy and along/cross position (K17).
    - Multiple data files may be read into computer memory using the LINK (K25) function
- Compare xy and along/cross track plots (K18, K19) to the navigation chart and COE data.
- If the plots do not agree with the navigation chart or COE data, calculate the offset in the waypoint that is necessary and use the Move (K24) function to reposition the waypoint.
- If the waypoint is "moved," recalculate xy and along/cross position data (K17), plot (K18, K19), and check plots against the navigation chart and COE data.
- When the waypoint is satisfactorily positioned, restore the waypoint table in the waypoint file (K23).

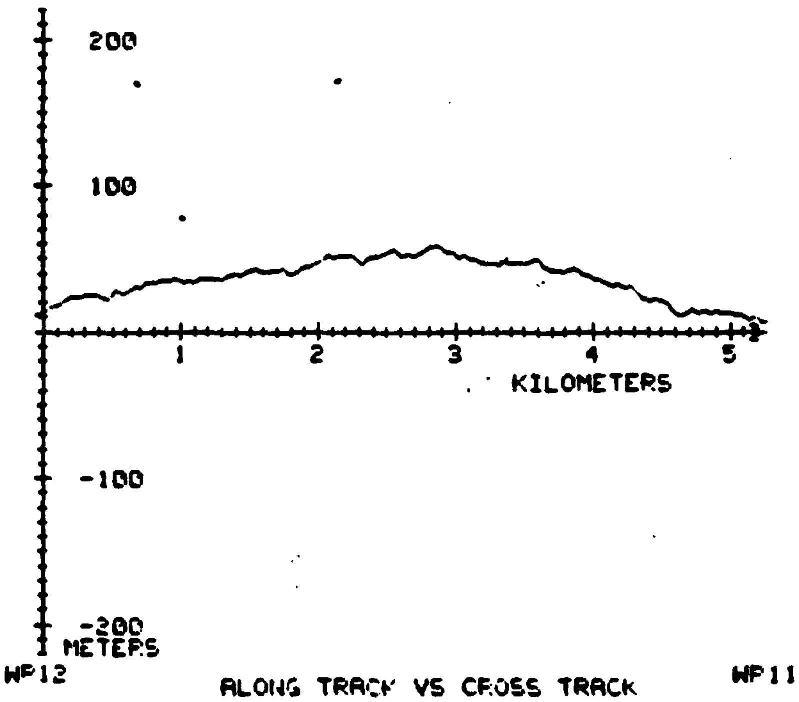
Figure 12
WAYPOINT CALCULATIONS - CHANNEL EDGE DATA

Waypoint Verification. If the waypoint was calculated using the channel edge data, the waypoint calculations are verified in the process of calculating the waypoint. The waypoint TDs are modified until position data converted from TDs for channel edges and aids-to-navigation agrees with the navigation chart.

In the case where the waypoint is defined by the intersection of two ranges, data should be collected along the entire length of the channel centerline during the survey. Data is usually also collected along channel edges and at several aids-to-navigation. Data collected along the channel centerline can be converted to xy and along/cross track coordinates using function K17. The plot of cross track vs along track (K19) is the best indicator of waypoint quality. If both waypoints are good and the vessel track was close to the centerline over the entire length of the channel, the cross track distance will be zero or near zero at both endpoints (e.g., at the waypoints). There may be a slight "bow" in cross track near the midpoint of the channel length. If the bow is severe a trackpoint can be inserted between waypoints. If the cross track distance is not near zero at one or both waypoints, recheck the waypoint calculations and check the survey notebook for comments as to whether the vessel was on range. It is good survey practice to collect data along the entire length of the channel several times during the survey. Several of these runs can be checked for consistent results before it is concluded that a waypoint is in error or a trackpoint is necessary. Data along channel edges and at aids-to-navigation can also be converted to xy and along/cross track position and plotted (K18 and K19). The plots can be compared to the navigation charts and COE data for agreement.

Another check of waypoint calculations is to calculate xy positions (using 3 TDs) for TD data collected in the vicinity of a waypoint and use the Project-Measured TD function (K27) to calculate the average TD fix triangle. The project function is the inverse of the Convert TD to XY and AT/CT function (K17). For each xy position, a set of TD values is calculated based on the distance from the waypoint position and the waypoint TDs. These calculated TDs are differenced from the measured TDs for each data sample. Statistics for the resultant differences are calculated and a summary table is printed on the hard copy printer.

Trackpoint Calculation. The need and location of a trackpoint can be determined from the along/cross track plot (K19) of a centerline data file. The trackpoint is generally placed where the "bow" in the plot is at a maximum. (See Figure 13.) The Move function (K24) in the range/bearing mode is first used to calculate the "ideal" TD and xy at the point on the trackline where the trackpoint is to be located. The changes in TD and xy calculated by K24 are applied to the waypoint values and stored in the waypoint table (K23) as the first estimates of TD and xy for the trackpoint. These trackpoint TD and xy coordinate values are simply an interpolation of the values at the two waypoints at each end of the trackline. To provide a correction for the "bow" in the along/cross track plot, the Move function is then used to offset the trackpoint in the crosstrack direction a distance equal to the magnitude of the "bow." The resultant TD and xy calculations are then stored in the trackpoint file. The trackpoint coordinates can then be used as the "to" waypoint for TD to xy calculations and the resultant along/cross positions replotted as a check of the trackpoint calculations. In some cases there may be a requirement for more than one trackpoint.



ALONG TRACK VS CROSS TRACK
WAYPOINT 11 TO WAYPOINT 12
FILE=C2022A

BOW IN CHANNEL CENTERLINE DATA

FIGURE 13

Additional trackpoints can be calculated as outlined above using the first trackpoint as one of the two waypoints. A sample worksheet for calculating trackpoints is shown in Figure 14. Rows 21-24 of the waypoint table are generally set aside for trackpoints.

Alternately, a trackpoint can be calculated using the range and bearing from the waypoint directly to the trackpoint and the MOVE functionto calculate the TDs in one step. The range (Rt) for the MOVE is

$$Rt = (R^2 + (bow)^2)^{1/2} \tag{31}$$

where

R = along track distance from the waypoint to the trackpoint

bow = magnitude of bow in centerline data at R

The bearing angle, B, for the MOVE is the sum of the course angle between waypoints (calculated from waypoint TDs) and the differential angle (dA) to the trackpoint

$$dA = \arctan (bow/R) \tag{32}$$

The course angle between waypoints (a) is found using functions (K17), convert TD Data to XY and AT/CT Position, and (K22), Daisy Chain. The bearing angle (B) is

$$B = A \pm dA \tag{33}$$

ELECTRONIC POSITIONING AUGMENTATION

The visual survey technique is difficult to apply in areas where there are no visual ranges, the channel boundaries are not distinct, and the channel is marked primarily with floating aids to navigation. However in such areas it is usually possible to provide electronic positioning coverage over a large area with a small number of reference stations. In such areas it becomes reasonable to use electronic positioning as a position reference for harbor Loran-C TD survey.

APPROACH

With electronic positioning available, the survey vessel should be stationed approximately at the waypoint and position and TD data recorded simultaneously. The position of the waypoint can be precalculated from COE survey data. Since the position of both the survey vessel and waypoint are known, the differential TD offset between the vessel and waypoint can be calculated accurately. This offset is calculated sample by sample, applied to the measured TDs, and the resultant TD data averaged to estimate the waypoint TD.

TRACK POINT CALCULATIONS:

•Trackpoint between WP ______ and WP ______

•Distance of Track point from WP ______, R= ______ km

•Bearing from WP ______TO WP _____, B= _____

•Results of "MOVE" of R&B from WP :

	dTDX =	dTDY =	dTDZ =
WP	TDX =	TDY =	TDZ =

First Estimate of Trackpoint = ______________ , ________________, ____________

WP____ Trackpoint XY Position X = Y =
dx = dy =

Trackpoint XY = ______________E , __________________N

•Trackpoint stored as WP ______;(K23)

•Magnitude of offset (e.g., BOW), M = ______ km

•MOVE trackpoint; R_M = M, B_M = $B \pm 90$ = ___________ (+ if bow to right)

•Results of "MOVE" of RM/BM from trackpoint estimate:

Estimate of Trackpoint = ______________ , ________________, ____________

WP____ Trackpoint XY Position X = Y =
dx = dy =

Trackpoint XY = ______________E , __________________N

•Recalculate xy and AT/CT position for verification data using trackpoint (K17)

•Replot AT/CT data (K19)

•If trackpoint satisfactory, restore waypoint table on waypoint file

TRACK POINT WORKSHEET

Figure 14

TD and electronic positioning data should be collected along the channel edges and centerline. Loran-C TD fixes can be compared to electronic positioning fixes to evaluate waypoint estimates and determine the need for trackpoints.

The use of electronic positioning simplifies shipboard data collection procedures and provides the capability to quantitatively evaluate the performance of Loran-C navigation using the calculated waypoints. It does however increase the complexity of the TDSS and infield logistics. There is also a significant increase in the time and effort necessary in the planning stage to prepare for the survey.

PRESURVEY PLANNING

Preplanning

The preplanning necessary for a harbor survey using electronic positioning is similar to the planning necessary for a visual survey. The survey strategy for each waypoint will be the same, i.e., station the survey vessel near the waypoint and record TD and positioning data. COE dredging data, e.g., coordinates of channel boundaries, is essential to this type survey. This data and the locations of reference stations which can be used to provide coverage of the survey area must be collected well in advance of the survey. The reference stations must be visited prior to the survey to insure that the survey markers have not been destroyed and that there is line-of-sight from the reference station to the survey area.

Waypoint Definition

The navigation chart for the area to be surveyed can be used initially to determine the location of waypoints. Where there are well defined channels, the intersections of channel centerlines are commonly defined as the waypoints. In an area where there is not a well-defined channel, local Coast Guard A-to-N personnel and users should be consulted to determine common vessel tracklines through the area. The intersections of these tracklines are defined as waypoints.

COE Dredging Data and Reference Stations

The COE dredging data define the channel boundaries in state-plane coordinates. Coordinates are also usually available for fixed aids to navigation in the survey area. These data and the reference station coordinates and descriptions are obtained from the COE office for the survey area. Figures 15 and 16 are examples of data for the Sandy Hook area of New York harbor.

The reference stations to be used for the survey area should be visited in advance of the data collection phase to insure that the survey markers still exist and there is line-of-sight from the station to the survey area. It is not uncommon for survey markers to have been destroyed since the last visit by the COE. In urban areas, new construction may obscure the line-of-sight from the reference station to the survey area. If the station has been destroyed or the line-of-sight obscured, an alternate station must be found.

The visit to the reference station also enables the survey party to plan

2 Feb. 1963 N.Y. HARBOR, LOWER BAY 15
Copy of: Chas. S VICINITY OF SANDY HOOK
INTERSECTIONS OF CHANNELS
DATUM: N.J.M.

NY, NJ, Bayside & Gedney Channels

Buoy Designation	Point	North	East	Line	Distance	Azimuth
8R	S^1	604 963.070	2 195 145.8[illegible]	$S^1 - S^2$	5782.3[illegible]	72° 09' [illegible].22
2R	S^2	603 007.606	2 189 632.204	$S^2 - S^3$	5635.613	77° 30' 20.[illegible]
4B New	S^3	601 812.033	2 182 203.710	$S^3 - S^4$	7619.209	67° 39' 02.71
3B New	S^4	599 663.411	2 175 129.095	$S^4 - S^5$	11,931.9[illegible]	105° 19' 26.51
Co.B.4	S^5	610 921.033	2 131 830.619			

Buoy Designation	Point	North	East	Line	Distance	Azimuth
6R	P^1	603 819.604	2 194 227.559	$P^1 - P^2$	5351.980	72° 09' 55.22
8R	P^2	602 233.109	2 188 994.870	$P^2 - P^3$	6896.952	77° 30' 20.17
1E New	P^3	600 805.900	2 182 444.121	$P^3 - P^4$	7357.055	67° 39' 02.71
3A	P^4	597 813.249	2 175 177.263	$P^4 - P^5$	3,312.016	105° 19' 26.51
3C 1A	P^5	598 557.782	2 171 809.704	$P^5 - P^6$	2,009.975	111° 32' 04.78
3C 2A	P^6	599 295.511	2 169 640.059	$P^6 - P^7$	33,711.926	105° 19' 26.51
Co.B.11	P^7	610 151.616	2 132 298.254			

South Channel

Buoy Designation	Point	North	East	Line	Distance	Azimuth
	P^5			$P^5 - K^1$	216.310	283° 07' 33.33
P1R	K^1	602 295.760	2 189,091.939	$K^1 - K^2$	23,926.813	307° 40' 52.11
	K^2	597,623.570	2 203,075.507	$K^2 - K^3$	800.000	37° 40' 52.11
	K^3	597 000.420	2 [illegible] 595.490	$K^3 - K^4$	21,632.574	127° 40' 52.11
P1R	K^4	600 223.377	2 190 455.097	$K^4 - K^5$	5,433.957	102° 33' 36.27
P2R	K^5	601 403.318	2 185 162.971	$K^5 - P^2$	2 812.[illegible]	287° 30' 20.17

Main Ship Channel

Buoy Designation	Point	North	East	Line	Distance	Azimuth
B'	B^1	615 490.910	2 174 065.911	$B^1 - E^1$	13,497.8[illegible]	89° 3[illegible]' [illegible]
E	E^1	602 036.360	2 172 763.426	$E^1 - E^2$	1,8[illegible].023	342° 32' 32.07
	E^2	600 263.063	2 173 218.944	$E^2 - E^3$	1,819.023	306° 35' 36.32
	E^3	599 160.635	2 174 703.292	$E^3 - E^4$	1,849.020	267° 13' 00.33
	E^4	599 247.781	2 176 550.971	$E^4 - S^4$	1,836.720	67° 39' 02.71
	S^4			$S^4 - D^4$	6 977.882	105° 19' 26.51
A	D^4	600 320.664	2 169 231.851	$D^4 - D^3$	1,066.074	210° 03' 33.19
	D^3	600 319.326	2 170 347.923	$D^3 - D^2$	1,066.074	335° 32' 01.32
	D^2	600 876.024	2 171 297.330	$D^2 - D^1$	1,066.074	210° 06' 34.81
D	D^1	601 423.045	2 171 741.330	$D^1 - B^2$	13 931.105	193° 32' 49.96
B2	B^2	615 297.173	2 173 073.329			

Channel to Navy Terminal at Leonardo

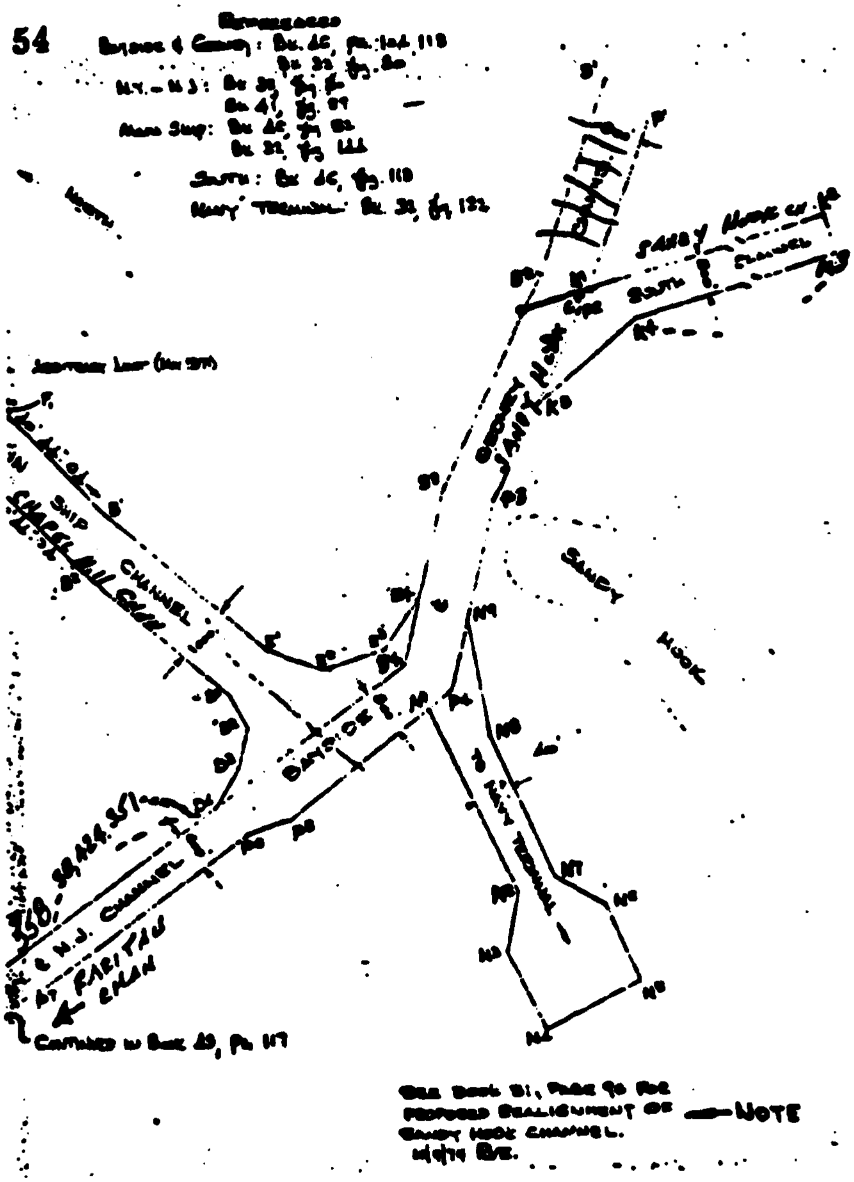
Buoy Designation	Point	North	East	Line	Distance	Azimuth
	P^4			$P^4 - N^1$	305.686	105° 19' 26.51
E	N^1	597,901.581	2 174 883.219	$N^1 - N^2$	3,063.263	87° 06' 30.16
2P	N^2	593 914.540	2 177 876.016	$N^2 - N^3$	1,319.517	61° 45' 10.86
2Q	N^3	592 819.265	2 171 908.250	$N^3 - N^4$	840.000	57° 06' 30.16
2R	N^4	592 093.641	2 171 117.986	$N^4 - N^5$	1,652.000	297° 06' 30.16
2S	N^5	591 323.926	2 172 801.743	$N^5 - N^6$	840.000	207° 06' 30.16
2T	N^6	593 080.160	2 173 149.067	$N^6 - N^7$	1,132.223	179° 08' 24.19
2U	N^7	593 212.270	2 172 932.014	$N^7 - N^8$	3,813.263	267° 06' 30.16
E	N^8	596 339.803	2 174 632.930	$N^8 - N^9$	2,903.842	230° 05' 16.73
	N^9	593 799.433	2 177 863.928	$N^9 - P^4$	2,830.312	67° 39' 02.71

Arbitrary Limit of Main Ship Channel (Designated as Chart-Hill No. [illegible] on [illegible])

Buoy Designation	Point	North	East	Line	Distance	Azimuth
	B_2			$B_2 - F_2$	10,022.406	1[illegible]° 40' 44.04
	F_2	625 441.007	2 174 933.925	$F_2 - F_1$	1,000.000	280° 40' 44.04
	F_1	625 295.703	2 175 910.206	$F_1 - B_1$	9,977.914	10° 40' 44.04

COE DREDGING DATA (SANDY HOOK)

FIGURE 15



COE DREDGING DATA (SANDY HOOK)

FIGURE 16

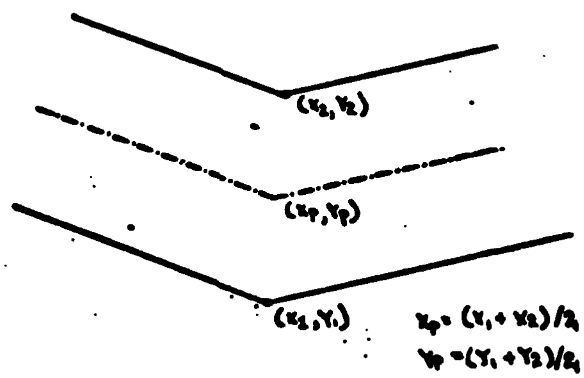
the logistics and personnel requirements for deploying the electronic positioning system transponders. Must the station be attended? Is there electrical power available? What is the travel time between reference stations?

<u>Coordinate Conversion and Waypoint Position</u>

To compare Loran-C position fixes and electronic positioning fixes, a common planar (i.e., xy) coordinate system must be used. A local origin with known state plane coordinates (or latitude, longitude) must be chosen at a central point in the survey area. If the entire harbor survey is done using electronic positioning, it is not necessary to be able to measure TDs at the local origin. The coordinates of the channel boundaries, fixed aids-to-navigation, and reference stations must be calculated relative to the local origin and the units converted from feet to kilometers.

The coordinates of waypoints are calculated based on the coordinates of the channel edges. There are two general approaches for calculating waypoint xy coordinates.

a. If the waypoint lies midway between two channel endpoints, the coordinates for the waypoints are simply the sum of the endpoint coordinates divided by two. See figure 17.

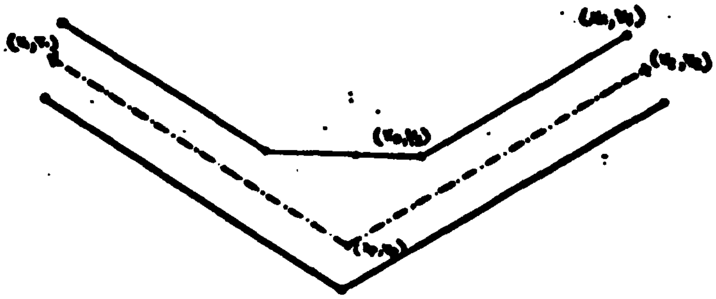


Straight Corner
Figure 17

b. If the waypoint is located at a cutoff corner (see figure 18), the waypoint must be calculated by using the channel edge data to write equations for the channel centerlines in the local coordinate system and then calculate the intersection of the two centerlines. The equation for the centerlines can be written in the following form:

$y-y_1 = M_1(x-x_1)$
$y-y_2 = M_2(x-x_2)$
where x_1, y_1, are the coordinates of any point on the centerline of channel one
x_2, y_2 are the coordinates of any point on the centerline of channel two
M_1 is the tan of the course angle for channel one
M_2 is the tan of the course angle for channel two

The course angles for the channel are usually listed in the COE data or can be readily calculated from two points on the channel edge, e.g., $M_2 = \tan((Y_4-Y_3)/(X_4-X_3))$ in figure 18.



Cutoff Corner

Figure 18

The coordinates of the waypoints in the local grid coordinates can be converted to latitude and longitude by first converting back to state-plane coordinates and then to latitude, longitude. The algorithms provided in Reference 3 permit conversion to state-plane coordinates from latitude, longitude and vice versa. The latitude and longitude of the waypoints are useful for predicting waypoint TDs and for describing the waypoint location to users.

Waypoint and Loran-C Chain Data Files

The data analysis program for a harbor Loran-C TD survey using electronic positioning augmentation, COMPAR, can be used during the the survey planning and data analysis phases. User instructions for COMPAR are contained in Appendix B. The program stores and uses three forms of base data files: Waypoint, Loran-C chain data, and Electronic positioning reference station. The Loran-C chain file contains the transmitter geodetic positions, transmitted power level and secondary emission delays. The waypoint files store waypoint TDs, latitude, longitude, and xy coordinates relative to the local origin. The waypoint file is a 25X8 matrix. Each row contains the data for one waypoint. Waypoint 25 is assigned to the local origin. The waypoints in the survey are divided into convenient sequences and numbered. A waypoint file is created for each sequence. The number of waypoints in each sequence is usually limited to twenty. This allows four spaces in the file for trackpoints if needed. The local origin is stored in the waypoint 25 position in each waypoint file. It is not necessary to be able to measure the TDs at the local origin for an EPA survey.

Use COMPAR, the Loran-C chain file, and geodetic positions of the waypoints to predict the TDs for each waypoint. The predicted TDs, calculated local xy coordinates, and geodetic coordinates of the waypoints can then be stored in waypoint files.

The reference station file must contain the state plane coordinates of the reference stations. The coordinates of the reference stations are used with the electronic positioning data to calculate vessel position.

Survey Plan Preparation

The aids to navigation at which the survey vessel will be stationed are selected from the navigation chart for each waypoint. If available, several aids which bracket the waypoint are chosen.

The TDSS can be used as a navigation aid when collecting electronic positioning data. If no aids are available near the waypoint, the TDSS can be used to station the vessel near the waypoint (preferred method) or maneuver in a cloverleaf pattern about the waypoint. It also provides cross track and along track position with respect to a trackline that is input in the form of start and stop points. Therefore, the TDSS can be used along the channel edges and the channel centerline (or any arbitrary trackline in the coverage area of the electronic positioning system). The end points of tracks that will be run to collect validation data can be recorded for use during the survey. Typically, verification runs are made along both channel edges and along the channel centerline.

DATA COLLECTION

Time Difference Survey System

The TDSS is designed to operate with the Mini-Ranger III Position Determining System. The HP9845 Desktop Computer and Mini-Ranger are connected via 16 bit and BCD interfaces. The 16 bit interface provides remote control of the Mini-Ranger; the BCD interface inputs range data.

Mini-Ranger.

The Mini-Ranger consists of a range console, control station receiver/transmitter unit, and two remote reference stations. The system operates in the 5400 to 5650 MHZ frequency band. Range resolution is one meter. Standard deviations of the range measurement vary with distance to the transponder (reference station). Typical values are 1 to 3 meters over the span of ranges of 5 km to 27 km. Each transponder unit can be individually calibrated to a known distance (see Mini-Ranger Calibration).

Graphics Display.

The TDSS provides two additional graphics display options when Mini-Ranger data is collected. Vessel position is calculated from the range data for each sample. The vessel position can be plotted on an xy or along/cross track scale. The xy display plots position relative to a waypoint or a point of interest (e.g., a fixed aid-to-navigation) entered in local coordinates. The operator also selects the area about the waypoint (or point of interest) that will be displayed. The along/cross track plot automatically scales the vertical height of the plotting area on the CRT to the distance between the start and stop points. The horizontal axes, cross track distance, is scaled to 200 meters either side of the trackline.

Mini-Ranger Calibration.

The Mini-Ranger is calibrated by adjusting measured distances for each transponder to agree with a known distance. Calibration is accomplished in the field using one of the methods below:

a. A calibration range is set up by marking off a distance in a clear area where the transponders, control station R/T unit, and range console can be set up. A distance of several hundred meters is sufficient for calibration. The disadvantages of this approach are that the range console and control station R/T unit must be removed from the survey vessel, and that an accurate distance must be measured independently of the Mini-Ranger. An advantage is that the calibration range can be set up in an area near the survey vessel that is easy to access, e.g., a large parking lot or a straight section of a lightly traveled roadway.

b. Use two existing survey markers as a calibration range. The disadvantages of this approach are that the equipment must be removed from the survey vessel and considerable time and travel may be involved to set up the transponders. Advantages are that a calibration range does not have to be measured and the survey markers can be chosen such that the calibration distance is the same order of magnitude as ranges expected during the survey.

c. Moor the vessel to a fixed aid to navigation with known coordinates in the survey area. Calculated distances between reference stations and the fixed aid are used as the calibration range. This is an ideal situation if such an aid exists and the transponders are visible from the vessel when it is moored to the aid. Care must be taken to measure offsets between the survey marker and control station R/T unit.

Periodic checks of the Mini-Ranger calibration should be conducted during the survey period. The calibration may be checked by using one of the above calibration approaches or:

a. if it is possible to cross the baseline between the two reference stations, the minimum sum of ranges should equal the baseline distance.

b. if there is an unsurveyed fixed aid or any location within view of the transponders where the survey vessel may be moored, ranges to the transponders can be measured shortly after calibration and recorded. The survey vessel periodically returns to this location to check that there is no change in the measured ranges.

c. if there is a visual range in the survey area, the survey vessel is centered on the range and cross track calculations are compared to visual observations. Care must be taken that the trackline start and stop points (e.g., waypoints) are entered correctly.

Waypoint Data

Loran-C TD and Mini-Ranger data are collected at aids-to-navigation near the waypoint. The survey vessel is usually moored to the aid as a matter of convenience. However, it is not necessary that the vessel be still in the water to collect data. If the sea state is such that it is not prudent to tie off to an aid, the data can be collected with the vessel stationed near the aid using it as a reference. Between 25 and 50 data samples are sufficient at each aid.

In the rare cases where there are no aids-to-navigation in the area of the waypoint, the TDSS is used as an aid to navigation to position the survey vessel near the waypoint. Some practice is necessary before using the TDSS to position the vessel since there is approximately a 10 second delay (depending on the sampling rate) after a sample is requested until the position is plotted on the graphics display.

Occasionally the path between the survey vessel and reference station is obscured by another vessel. The TDSS is designed to time out if a sample is not received from the Miniranger within three seconds of a data request. A "TIME OUT ERROR" warning statement is printed on the hard copy printer any time a time out occurs and TD data is not collected.

If the range data received from the Mini-Ranger is not within 1 km of the previous sample, the data is rejected and a "RANGE ERROR" is printed. This feature automatically edits most multipath data. Occasionally a bad range sample will pass the range test. These samples are easily detectable on the xy or along/cross track plot. When such errors are observed, the sample numbers should be recorded for later editing.

Waypoint Verification Data

Data to be used for waypoint verification and trackpoint calculation if necessary is collected along channel edges and centerlines. The TDSS can be used to provide near realtime along/cross track position information for both channel edge and channel centerline data collection runs. It is not necessary to maintain the vessel track accurately along the channel boundary or centerline. Since position data is collected, a sample by sample comparison of Loran-C position fixes using calculated waypoints and Mini-Ranger position fixes is possible. Data should be collected nominally along channel edges and channel centerlines to evaluate the performance of Loran-C navigation over the entire channel.

DATA ANALYSIS

COMPAR

A Hewlett Packard HP9845 desktop computer and the program, COMPAR, are used to analyze the data collected during a harbor survey using electronic positioning augmentation. The program, COMPAR, is a collection of special function programs which can be used to:

a. calculate waypoint TDs and
b. evaluate the performance of the resultant waypoints.

The program COMPAR includes 19 function programs selected by special function keys. The functions with short descriptions are divided into five groups below.

Presurvey Planning Group

a. Predict TDs, K20: This key calculates predicted TDs for an input latitude, longitude based upon an all-sea water earth. The program also outputs range and bearing to transmitter stations and GDOPs for the three TD and two TD fixes.

b. File or Read Waypoint Data, K23: This key is used to create waypoint table files and to edit and restore waypoint table files. It can also be used to obtain a listing of the waypoint table. The waypoint table is a 25X8 matrix. The first four columns contain waypoint TD values (TDW, TDX, TDY, TDZ). Columns five and six contain the xy position referenced to the local origin. Columns seven and eight contain the latitude and longitude in decimal degrees. Waypoint 25 (e.g. row 25) is designated as the local origin. The latitude and longitude stored in this location are used by any program function which calculates the local xy coordinates of the transmitters. If the local origin is also one of a sequence of waypoints, its parameters will be stored twice, i.e. in row 25 and in the row corresponding to its logical waypoint number.

c. Store Chain Data, K26: This key stores Loran-C chain data (transmitter geodetic positions, transmitter power levels, and secondary emission delays) on a data file for later use. File names are a five character neumonic for the particular Loran-C chain. The first four characters are an abbreviation for the chain (e.g. NEUS, SEUS, GTLK, etc.). The fifth character is a number from 1 to 4 which designates the configuration of three TDs: 1 = XYZ, 2 = WXY,
3 = WXZ, 4 = WYZ.

d. Store Reference Station Data, K28: This key is used to store reference station coordinates on a file for use in converting range data to xy coordinates.

Data File Handling Group

a. Read Data File, K0: This key reads time-difference (TD), range and time-of-day data stored on magnetic tape. The range and TD data may be corrected for known errors. If the range data is corrected, a lower case "r" is annotated to the file name, F$. If TDs are corrected, a lower case "t" is annotated to F$.

b. Edit Data, K16: This key is used to edit (i.e. remove) samples from the data arrays. Four options are available to the user:

(1) The first option deletes a single data sample in each data array.
(2) The second deletes a block of data samples.
(3) The third deletes samples with TD samples outside a range that is input by the operator.
(4) The fourth deletes samples with outliers in the range data. The routine compares each range with the average of the previous and next sample. If there is a difference of more than 100 meters, the sample is deleted.

Unfortunately, this technique usually deletes the outlier and the sample on each side of the outlier. This is normally not a significant problem, i.e. nine samples will be deleted rather than three. The alternative is to detect the outliers and the corresponding sample number using the Plot XY Position function (K18) and use the single sample delete option.

When editing of the data arrays is complete, the operator may store the edited data on a new file for later use.

NOTE: The edit function deletes only data samples. Arrays containing XY and along/cross track position data are not affected. Functions K1 and K17 must be repeated to reflect the editing on computed positions.

c. Link Data Files, K25: enables multiple data files to be loaded into memory. Range and TD data can be corrected for each file entered. The total number of samples must be equal to or less than 400. The function will automatically ignore any samples which would cause this limit to be exceeded.

Statistics and Regression Group

a. Calculate Statistics and Linear Regression of TD Data, K2: This key calculates statistics, linear regression coefficients, and minimum and maximum for the four TD arrays. Statistics ahd regression parameters calculated are:

(1) mean for each TD
(2) standard deviation for each TD
(3) covariance of TD pairs
(4) correlation coefficient of each TD pair
(5) linear regression slopes of each TD pair
(6) standard deviation of residuals for each regression line

b. Plot TD Data with Regression Lines, K4: This key plots two of the time-difference arrays against each other and also plots the linear regression line for the TD pair chosen. The program automatically scales the plot for the minimum and maximum TDs for each array. The axes are drawn through the mean for each TD. Minor tic marks are every microsecond and major tic marks every 10 microseconds.

c. Plot Residuals, K5: This key plots the residuals from the linear regression of any TD pair (See K2). The residuals may be plotted against sample number or the independent variable. Residuals are normalized to the standard deviation of the residuals. Normalized values greater than 5 are printed on the hard copy printer and are not plotted.

Position Calculation Group

a. Convert Mini-Ranger Data to XY and Along/Cross Track Position, K1: This key converts Miniranger range data to cartesian xy coordinates relative to a local origin. Cross track and along track data are also calculated. Trilateration is used to calculate xy position using distances measured to two reference stations and the distance between reference stations. Cross track and along track position are calculated relative to a trackline between two waypoints.

b. Convert Loran-C Data to XY and Along/Cross Track Position, K17: This key converts TD data to xy and along/cross track positions using the FEHG algorithm and surveyed waypoints. The program will compute either a three or two TD solution. A summary table is printed on the hard copy printer which lists:

-chain and LOPs used in the solution
-file name
-bearing angle between waypoints used for along/cross track calculation
-rms trackline of data
-average cross track position
-standard deviation of cross track position
-average xy position
-standard deviation of xy position data

c. Plot XY Position, K18: This key plots the xy data calculated from Loran-C TDs and/or Mini-ranger data. Four options are available:

(1) Mini-ranger data only
(2) Loran-C data only
(3) Mini-ranger and Loran-C data
(4) Error plot

The first two options allow the operator to blow-up a section of the plot, to find the sample of numbers of outliers, and to digitize up to 10 locations on the plot. A hard copy option is provided with all four plots. The "Error plot" plots a vector at each sample. The tail of the vector is the mini-ranger position fix; the head (denoted by a "o") is the Loran-C position fix.

d. File or Read Waypoint Data, K23: See Presurvey Planning

e. Plot Along/Cross Track Data, K19: This key plots cross track vs along track position. Three options are available:

(1) Mini-ranger data
(2) Loran-C data
(3) Mini-ranger and Loran-C data

Waypoint Evaluation Group

a. Compare Mini-Ranger and Loran-C Position Data, K3: This key compares Loran-C derived position with miniranger calculated position. Mini-ranger positions are assumed to be "truth" and the difference between the Loran-C position and Mini-ranger position is the Loran-C position error. Two sets of position errors are calculated, XY and along/cross track. The XY errors are a direct comparison between Mini-ranger and Loran-C derived positions. The Miniranger along/cross track position is based on the local coordinates of the waypoints. The Loran-C along/cross track position is relative to the course line calculated from the difference in TDs between waypoints.

The following parameters are calculated:

(1) average x and y position error
(2) rms x and y position error
(3) average along and cross track error
(4) rms along and cross track error
(5) rms radial error based on xy position errors
(6) rms radial error base on along/cross track errors

b. Calculate TD Grid Warp, K21: This key calculates the difference between measured TDs and TDs projected from a nearby waypoint. This difference is termed "TD grid warp" and is basically a measure of the change in Additional Secondary Phase Factor (ASF) over the area where data was collected. TDs are projected from the waypoint to the measurement point based on the difference between the measured and waypoint positions. There is a small error in the calculation due to use of a flat earth model for transmitter locations. This error is typically less than 40 nanosec within 12KM of the waypoint.

c. Plot XY Position, K18: See Position Calculation Group.

d. Plot Along/Cross Track Data, K19: See Position Calculation Group.

Waypoint Calculations

The primary method to calculate waypoint TDs is to "reflect" TDs collected near a waypoint to the waypoint. The "reflect" function (see K27) calculates the difference in TDs between the measurement position and the waypoint position. This difference is applied as a correction to the measured TDs for each sample. The mean and standard deviation of the "reflected" TDs are then calculated.

The standard deviation of the reflected TDs are normally slightly larger than the standard deviations of data collected at dockside. Outliers in the Mini-ranger data are the usual cause of high standard deviations in the reflected TD data. It is a good practice to edit the data for range errors (K16, option 4) prior to reflecting the data to the waypoint.

Data can be collected at several points near a waypoint (typically moored to or stationed near a buoy in the vicinity of the waypoint). An optional data collection technique is to maneuver the survey vessel in a cloverleaf pattern centered at the waypoint. Waypoint TD estimates are calculated from each data file. The results are compared to each as a check for errors. Agreement to within 20 to 30 nanoseconds of the mean can be expected. Differences are the results of estimation errors due to noise, uncorrected TD offsets, the difference in distance between the measurement positions and the waypoint, uncorrected range errors, etc. It is the judgement of the data analyst how to combine the waypoint TD estimates for each data file to form the final TD estimate. The most straightforward approach is to average the estimates. Figure 19 is a flow chart for calculating waypoints TDs.

Note that the "Daisy Chain" function is not used to calculate waypoint xy coordinates. The COE based coordinates are retained in the waypoint file. The "Daisy Chain" function is used to check that the range and bearing calculated between waypoints using the difference in TDs is approximately the same as calculated using the COE based coordinates.

.For each data file collected near the waypoint of interest

-Read data file (K0)

Correct range data
Correct TD data

-Edit data for range errors (K16, option 4)

-Convert range data to xy position (K1)

-Reflect TDs to waypoint of interest (K27)

.Calculate final waypoint estimate (i.e. average results from each data file)

.Store waypoint TDs in waypoint file (K23)

.Do <u>not</u> change xy position of waypoint

FLOWCHART OF WAYPOINT CALCULATIONS, EPA SURVEY

FIGURE 19

Waypoint Verification

The TD and Mini-Ranger data files collected along channel edges and centerlines can be used to compare the position fixes obtained from Mini-ranger and the positions calculated from the TDs using calculated waypoints. This comparison provides a verification of the waypoint calculations, provides a measure of performance which may be expected using a Loran-C navigation device (e.g., PILOT), and enables the data analyst to decide if a trackpoint is necessary between waypoints.

The coordinate system for waypoints and Mini-ranger reference stations are based on Army Corps of Engineers (COE) dredging data. The COE coordinates are in state plane coordinates. These coordinates can be translated to local coordinates by subtracting the state plane coordinates of the local origin and changing units from feet to kilometers. Waypoints are calculated as the intersection of channel centerlines. (See Presurvey Planning.)

The position fixes for Mini-ranger data are computed using trilateration. Along and crosstrack positions are based on the range and bearing calculated between tabulated waypoint coordinates.

The Flat Earth Hyperbolic Grid (FEHG) algorithm is used to compute xy coordinates from TD data (K17). Inputs to the algorithm are calculated waypoint TDs, waypoint xy coordinates and transmitter xy coordinates. The transmitter xy coordinates are referenced to the local origin. The FEHG algorithm is also used to calculate the coordinates of the next waypoint for along/cross track calculations.

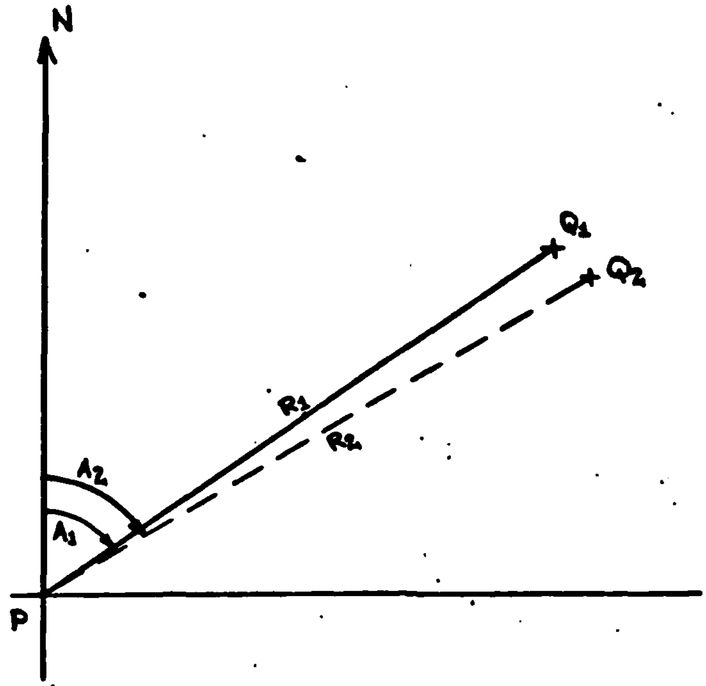
The function "Compare Mini-ranger and Loran-C Position Data" (K3) calculates the difference in xy position and along/cross track position for each data sample and the mean and rms values of the differences for the data set. If the Mini-ranger positions are assumed to be "truth," the result is an evaluation of the accuracy of the Loran-C fixes for the data set. Plots are provided for the xy "errors" and along/cross track "errors." Plots of the xy data and along/cross track data on the same axes are also available (K18 and K19).

One must be careful to compare "oranges to oranges" when evaluating the "error" statistics and plots generated by the function K3. The domain in which a Loran-C harbor navigation device (e.g. PILOT) operates is distances relative to a trackline defined by two waypoints (i.e. along/cross track). The xy "error" data is interesting, but its only significance is that the x and y errors should approach zero near the "TO" waypoint (along track distance = 0). Large xy errors near the "TO" waypoint is indicative of an error in the waypoint, the Mini-ranger data, or the Loran-C data.

The along/cross track "errors" are the more significant. If both the waypoint "TO" and "FROM" waypoint calculations are correct, the cross track distance "errors" at each end of the trackline will be near zero. The along track distance errors should converge to zero at the "TO" waypoint. Note that for the Mini-ranger data, along and cross track distance are calculated using the distance and bearing calculated between the waypoint coordinates determined from COE dredging data (see Figure 20). For the Loran-C data, the "TO" waypoint xy coordinates are the same as those used in the Mini-Ranger calculations. The "FROM" waypoint coordinates (Q_2 in Figure 20) are calculated using the FEHG algorithm and the waypoint TDs. The range and bearing between P and Q_2 (R_2 and A_2) are not, in general, equal to the range and bearing between P and Q_1 (R_1 and A_1).

Trackpoint Calculations

Cross track error is the best measure of the expected performance of Loran-C navigation between waypoints. The plot of the cross track error may exhibit a "bow" between the waypoints. That is, the error will be near zero at both endpoints, and there will be a well defined offset somewhere near the halfway point. The severity of this bow and the channel dimensions are the factors which must be evaluated to determine if a track point is necessary. A trackpoint is basically a waypoint located on the centerline between two waypoints. The data analyst determines the location of a trackpoint(s) by inspecting the cross track error plot. The along track distance where a trackpoint(s) is needed is recorded. The function, Calculate Grid Warp (see K21), is used to calculate the difference between the measured TDs and the TDs projected from the waypoint based on the difference in position between the waypoint and measurement point. The differences are plotted against along track distance. The correction to be applied at the trackpoint for each TD can be picked off the error plot at the along track distance the trackpoint is to be located. The ideal TD at the trackpoint is calculated using the TD Move (K24) function. The trackpoint TD is equal to the ideal TD plus the correction determined from the TD error plot. A sample worksheet for calculating trackpoints is shown in Figure 22.



P - Waypoint "TO" xy position; calculated from COE dredging data; stored in waypoint table

Q_1 - Waypoint "FROM" xy position; calculated from COE dredging data; stored in waypoint table

Q_2 - Waypoint "FROM" xy position calculated using FEHG algorithm referenced to P

A_1 - bearing from P to Q_1

A_2 - bearing from P to Q_2

R_1 - range from P to Q_1

R_2 - range from P to Q_2

FIGURE 20

.Read data file (K0)

Correct range data
Correct TD data

.Edit data for range errors (K16, option 4)

.Convert range data to xy position (K1)

.Convert TD data to xy position (K17)
(Use same "TO" and "FROM" waypoints as above)

.Compare the two sets of xy data (K3)

.Plot xy data (K18) - optional

.Plot along/cross track data (K19) - optional

.If cross track errors are significant

Calculate grid warp (K21)
Calculate track point

SUMMARY OF PROCEDURE TO COMPARE POSITION DATA
CALCULATED FROM MINI-RANGER AND LORAN-C

FIGURE 21

TRACKPOINT CALCULATIONS (EPA SURVEY)

.Trackpoint between WP____ and WP____

.Trackpoint location; from WP____
Range = ________
Bearing = ________

.For "MOVE" of R, B from WP_______

dTDW = ____________ dTDX ____________ dTDY ____________ dTDZ ____________
dX = ____________ dY = ____________

.For WP_______

TDW ____________ TDX ____________ TDY ____________ TDZ ____________
X= ____________ Y= ____________

.TD(WP___) + dTD = FIRST ESTIMATE OF TRACKPOINT

TDW = ____________ TDX = ____________ TDY = ____________ TDZ = ____________

.For TD "WARP" Plot; TD corrections are:

cTDW = ____________ cTDX = ____________ cTDY = ____________ cTDZ = ____________

.Trackpoint TD = FIRST ESTIMATE + CORRECTION

TDW = ____________ TDX = ____________ TDY = ____________ TDZ = ____________

.Trackpoint XY

X(TP) = X(WP___) + dX Y(TP) = Y(WP___) + dY
X(TP) = ________________ Y(TP) = ____________

TRACKPOINT CALCULATION WORKSHEET

FIGURE 22

PERSONNEL, SCHEDULING AND TRAINING

PERSONNEL

The minimum recommended survey crew for a Visual Reference Survey is three. Two of the survey crew collect data aboard the survey vessel; the other remains behind to analyze data and provide necessary support. These duties should be rotated during the survey. When Mini-Ranger is used for a position reference, the survey crew increases to four or five. One or two additional personnel are required to setup and, if necessary, man the reference station units.

SCHEDULING

A typical harbor area with 20-25 waypoints will require less than two weeks to survey. For the test surveys of New York Harbor and Delaware Bay and River the survey equipment was installed and tested on a Monday; data was collected Tuesday through Friday and Monday through Wednesday of the following week; the weekend was used for preliminary data analysis. The equipment was removed from the survey vessel in approximately three hours. A typical survey day is 10-14 hours from equipment turn-on in the morning to the final set of dockside reading upon return.

The New York (EPA) survey required several visits to the New York area to visit the COE, locate reference stations, and become familiar with the area. For a Visual Reference Survey a presurvey visit to the area to be surveyed is recommended. For an EPA survey, the presurvey visit is essential.

TRAINING

The members of the survey team should be familiar and/or receive training in the following areas:

a. Hewlett Packard HP9845 Desktop Computer

b. Austron 5000 Loran-C Receiver

c. Motorola Mini-Ranger

d. Basic Statistics

e. Linear Regression Theory

f. Coordinate conversion algorithms, particularly the FEHG algorithm used in PILOT equipment

g. State Plane Coordinates

SURVEY VESSEL

The ideal survey vessel is 30-40 feet long with a draft of 2-3 feet. It has twin screw or stern drive propulsion and AC generator capacity of approximately 5 kilowatts. Maximum speed is 20-25 knots. There should be both interior and flying bridge controls. The cabin area should provide a large work area which includes a chart table, space for the TDSS, and a desk for data analysis. Electronic equipment should include a fathometer and a VHF FM radio.

Unfortunately, such a vessel is not currently in the Coast Guard fleet. A 65 foot harbor tug (WYTL) was used successfully to conduct harbor surveys in New York Harbor and Delaware Bay and River. There is enough space in the pilot house for the HP9845 computer and Mini-Ranger control unit. In one survey the Austron 5000 was located on the mess deck and in the CPO Quarters for the other. In both cases the TDSS was operated without the UPS, and no difficulties related to electrical power were experienced.

A commercial vessel was used for each of the surveys of the St. Marys River. For the survey of the Mini-chain, a 33 foot Egg Harbor was the survey vessel. There was sufficient working space in the main cabin when all the furniture was removed. A 38 foot houseboat was used for the survey of the Great Lakes Chain. The houseboat provided ideal working space, but was not as sea worthy as the Egg Harbor. In both of the St. Marys River surveys, the survey crew operated the survey vessel. This provided increased flexibility for scheduling survey operations, but is more demanding of the survey crew.

IMPROVEMENTS

The sections below describe areas for improvement and expanded capability (and utility) of the TDSS. Analysis software development to support the expanded utility of the TDSS is also described. An area for follow on development work is discussed in SURVEY TECHNIQUES.

TDSS

Design changes to the TDSS software were frozen in November 1980. Several of the following changes were not considered essential to the TDSS handoff.

Initialization

In the current version of the TDSS software, the parameters for the TD graphics display (origin, minimum and maximum limits) must be inserted manually. A revision of the data collection initialization software to minimize the number of manual entries will simplify the setup for data collection during a visual reference survey. This revision can be implemented by reading the waypoint file (with predicted or measured data) into computer memory. The operator would then enter waypoint number rather than TD values. Plot limits could be input relative to the waypoint (e.g., -1.0, +2 microseconds for X-axis; -3, +2 microseconds for Y-axis)

On-Line TD to XY Coordinate Conversion

Verification data is processed during the data analysis phase of the survey. If an on-line TD-XY coordinate conversion capability existed within the TDSS, the positions fixes (along/cross track) could be compared to a visual range as check of calculated waypoints and the requirement for a trackpoint as the data was collected. The conversion algorithm subroutines in TLS1 and COMPAR (Fehg and Fehgt) can be easily added to the TDSS software. The along/cross track and xy plots used during EPA survey can also be used to plot position data calculated from TD data. The addition of on-line TD to xy conversion will also provide the capability to use the TDSS to demonstrate Loran-C positioning in harbor areas before a set of PILOT tapes is developed.

Data Storage

Additional information stored on the data tape would assist data analysis. The provision for a comments section to be stored with each data file would allow notes taken by the survey party to be recorded with the data on magnetic tape. The data analyst would then be provided with a short description of the data set, samples to be edited, etc., when the tape was read into the HP9845.

When reducing the data collected during an EPA survey, the analyst must refer to the survey notebook for the file names which contain the reference station data used for the survey data run. He must also enter which station the "R1" transponder was located. This data could be stored as part of each data file and input with the TD, range, and time data.

Loran-C Receiver

The Austron-5000 Loran-C Receiver is a heavy and bulky piece of electronics equipment. It is also more complicated to operate than most modern Loran-C receivers. A compact modern Loran-C receiver with 10 nanosecond resolution (e.g., Internav 404) would simplify transportation, survey vessel installation, and operation of the TDSS.

The advantages of the Austron-5000 are that its performance is well known; it is standard Coast Guard equipment; it has a fixed multi-notch filter chassis; and it is more versatile than most Loran-C receivers designed for commercial use.

Expand Positioning System Capability

Motorola Mini-Ranger was chosen as the electronic position reference for the TDSS because it met the performance requirements and a set of equipment was on hand during the TDSS development. Other systems (e.g., Cubic AUTOTAPE) are capable of providing satisfactory position data for EPA surveys. The TDSS can have a wider range of application if the software is also expanded to interface to longer range positioning systems such as RAYDIST, ARGO, etc. It could then be used for chart verification in CCZ areas as well as harbor surveys.

Modified TDSS

A prototype TDSS designed to use an Internav 404 Loran-C receiver has been designed and implemented. It uses the same basic data collection routines as the Austron 5000 TDSS but dispenses with the receiver control portion of the program. The HP 9845 calculator receives a line of TD data from the COMMS port of the receiver in much the same way as is employed in the PILOT user equipment. A TTL to EIA level shifter is employed between the receiver COMMS port (TTL) and the RS-232 serial interface on the HP 9845.

The Internav 404 TDSS utilizes special function keys 1 through 7. The functions are displayed at program initialization and are listed below.

K1	PAUSE
K2	CONTINUE
K3	DATA COLLECT
K4	START
K5	STOP
K6	STATS
K7	SIGN

These functions are identical to those described in table 2.

The LC 404 TDSS program entitled 404S was successfully used to verify a visual survey of the St. Laurence River. The EPA portion of the program was revised and successfully implemented in a survey of the Strait of Juan de Fuca (Washington state). Unlike the Austron 5000, the tracking loop time constant on the LC 404 receiver is fixed at approximately 8 seconds. Thus it is not as responsive to changes in velocity. Care must be taken by the survey party to smoothly maneuver, accelerate and decelerate the survey vessel. To insure independent samples, a sample period of no less than 15 seconds should be selected during data collection.

DATA ANALYSIS SOFTWARE

The programs TLS1 and COMPAR have grown near the limits of available memory in the HP9845. The addition of a function to either program will probably require that an existing function and its associated subprograms be deleted. Care must be taken when deleting a subprogram since it may be used by several program functions. Appendices A and B each contain a list of their subprograms and the program functions where they are used. Another note of caution is that subprograms in TLS1 and COMPAR with the same name may not be identical, particularly plot subprograms.

Expand TD to Position Coordinate Conversion Capability

The TDSS can be used to demonstrate Loran-C navigation and/or evaluate TD to position coordinate conversion algorithms with the addition of the conversion software in the form of subprograms. The design of the TDSS software allows such expansion to be added without significant changes to the basic program structure. The changes are equivalent to changing a printed circuit board module in electronics equipment.

Expand "Grid Warp"

The function calculate Grid Warp in COMPAR calculates the difference between predicted TDs and measured TDs for a data set. The predicted TDs are about a known reference point based on the difference in distances to the transmitters between the measured data position and the reference point and measured TDs at the reference point. If the additional secondary phase factor, ASF, is constant in the area about the reference point, the measured and predicted TDs will agree. Changes in ASF over the area will be seen as "grid warp." This function can be expanded to analyze data collected over a large area in the CCZ to determine the behavior of ASF by calculating predicted TDs minus measured TDs. Predicted TDs are calculated using standard techniques (e.g., EEE-10) and position data collected from a long range positioning system.

Develop Self-Calibrate Software

The concept of user self-calibration is discussed in the following section. Analysis software to demonstrate this approach must be developed, while the TDSS could be used without modification to collect data for this demonstration.

SURVEY TECHNIQUES

Self-Calibration

The concept of user self-calibration is that Loran-C user equipment can be designed to collect data on several trips into and out of a harbor area and this data then used to calculate an average inbound and outbound trackline. The inbound and outbound tracklines can be divided into convenient segments for display on a CRT. A straight line or curve must be fit to the data. The user equipment can plot the average trackline and ship position calculated from measured TDs. Alphanumerics on the display can indicate cross track distance, cross track speed, and speed over ground. The approach is similar to a track plotter. A track plotter could be used to plot inbound and outbound tracklines for a harbor. The plot would then be saved until the next visit by the vessel to the harbor. With the pen in the up position, current ship position indicated by the point of the pen could be compared to the trackline plotted previously. A limitation of this approach is the scaling necessary to display the entire harbor area on the plotting surface or the necessity to change paper several times during a transit. This technique would also not provide a means for calculating cross track distance.

Demonstration of the user self-calibration technique can remove the Coast Guard from the harbor survey business. It would be up to each manufacturer to develop a specific calibration technique for its equipment.

REFERENCES

1. Olsen, D. L.; Ligon, J. M.; Sedlock, A. J.; and Isgett, C. E., PRECISION LORAN-C NAVIGATION FOR THE HARBOR AND HARBOR ENTRANCE AREA, May 1980. Published as U. S. Coast Guard Report No. CG-D-32-80. Available from National Technical Information Service, Springfield, VA 22161. The accession number is AD-A086001.

2. Milne, T. J.; Sedlock, A. J.; and Lewicki, J. P., TIME DIFFERENCE SURVEY SYSTEM (TDSS). Not yet published.

3. Claire, C. N., STATE PLANE COORDINATES by AUTOMATIC DATA PROCESSING, U.S. Department of Commerce Publication 62-4.

APPENDIX A

TLS1, Data Analysis Software for Visual Reference Survey

DOCUMENTATION FOR PROGRAM "TLS1"

TABLE OF CONTENTS

PROGRAM APPLICATIONS, "TLS1"

The program "TLS1" is a collection of special function programs used to analyze data collected during a Loran-C Harbor Survey using visual aids to navigation and channel features as a position reference. The major functions of the program are:

a. calculate waypoint Loran-C time differences

b. evaluate the performance of the resultant waypoints

The following is a set of general procedures to use in applying TLS1 to analyze data collected during a harbor survey using visual aids to navigation as the position reference.

PRELIMINARIES. Waypoint table and Loran-C chain data, data files are necessary for complete data analysis. The waypoint table file contains time-differences, local xy coordinates, latitude and longitude for twenty-five waypoints (see K23). The Loran-C chain data file contains latitude and longitude and power level for three secondaries and the master and the emission delays for the three secondaries (see K26). The waypoint table and Loran-C chain data data files are prepared during the survey planning stages. It is convenient to have the waypoint table file(s) and chain data file on the same data tape cartridge.

The Loran-C TD data may be corrected for known offsets (see K0) during the time period that data was collected. Offsets can be determined from a local monitor TD record and data from the system area monitor.

WAYPOINT CALCULATIONS. Waypoint calculations are divided into two categories:

a. waypoints defined by the intersection of two channel centerlines which are both marked by visual ranges.

b. waypoints defined by the intersection of two channel centerlines which one or both are not marked by a visual range.

Visual Range Solution:

For the case where the waypoint is defined by the intersection of two visual ranges, data will have been collected on each range line and stored in a data file. The term, trackline, is used for the survey line along one of the visual ranges. Tracklines are run near the waypoint and optimally bracket the waypoint. Typically a trackline data file contains three to four runs along the trackline. For each pair of TDs collected, a straight line is fitted to the trackline data. The intersection of the lines fitted to each trackline is an estimate of the waypoint TDs. The function, Compute Waypoint (K21), performs the above calculations. Figure 1 is a flow chart of the procedure to be used to calculate waypoint TDs for a waypoint defined by the intersection of two visual ranges.

The function, Compute Waypoint (K21), calculates statistics and linear regression parameters for each TD pair of data collected for the two trackline data files and the intersection of the regression lines for each TD pair. The crossing angle of the regression lines and the estimated rms error of the waypoint estimate (intersection of regression lines) are calculated and the results are printed in tabular form.

The quality of the trackline data can be checked by examining the correlation coefficient and residuals for each TD pair. Four TD pairs can be collected, but it is rare that all four provide useful data. In most cases, there are three good TDs available The correlation coefficient and residuals are a measure of how good the trackline data fits a straight line. If the vessel track were a perfect straight line, the residuals would be slightly greater than the standard deviations observed at dockside. A "noisy" vessel track will show up in larger residuals. A typical range in residuals is .025 to .060 microseconds. One's ability to detect if the survey vessel is on or off the range decreases as distance to the range markers increase. Therefore, tracklines run at the far end of the range will tend to have larger residuals than tracklines nearer the range markers. In some cases, portions of tracklines run with the range markers "over the shoulder" may cause problems due to poor track keeping. The survey officers notebook should be checked for comments concerning suspect data. Suspect data can be removed using the Edit (K16) function. TD Statistics and Regression (K2), Plot TD Data with Regression Lines (K4) and Plot Residuals (K5) are useful functions for detecting suspect data and identifying sample numbers to delete.

The estimated rms error and the closure between waypoint solutions is a good indication of waypoint quality. Estimated rms error is a function of trackline residuals, crossing angles, number of samples, and the survey pattern. Reducing trackline residuals by editing suspect data will reduce rms error in the waypoint solution. Crossing angles are fixed by the configuration of the channel and the Loran-C chain geometry. The rms error is inversely proportional to the square root of the number of samples. The error is minimum when the survey tracklines form an X pattern. The difference between an X pattern and a V pattern is a factor by two. Editing data to better bracket the waypoint can improve the calculated rms error. Note: it is not a good practice to edit data such that the trackline does not reach the waypoint.

If there are three TD data arrays, there will be two solutions for the waypoint value of each TD. If four TDs are collected, there will be three solutions for each waypoint TD. The term, closure, is used to describe the agreement between solutions. If two solutions have low estimated rms errors (e.g., less than 20 nanoseconds), there should be good closure between the solutions.

The waypoint TD estimate with the lowest estimated rms error should be chosen as the waypoint value. If two solutions have approximately the same rms error, they may be averaged.

•Calculate intersection of survey tracklines (K21)

•Check quality of trackline data and waypoint solution.
- •correlation coefficients (greater than .95)
- •trackline residuals (20-60ns)
- •estimated rms of waypoint solutions
- •closure of waypoint solutions
- •crossing angles

•If data not satisfactory, edit and restore data files, and recompute waypoint.

•If results still not satisfactory, resurvey waypoint.

•Choose waypoint estimated with lowest estimated rms error as waypoint value.

•Store TD data in waypoint file.

•Calculate and store waypoint xy in waypoint file.

Waypoint Calculation -Visual Range Solution

Figure 1

Non-Visual Range Solution:

An iterative approach has been developed for the situations where there is only one or no visual range available. In this case, TD data is collected along channel edges, the channel centerline (if a range is available), and at aids-to-navigation near the waypoint and along channel edges. The TDs of a fixed aid near the waypoint (a buoy may be used if it is the only aid available) are stored temporarily as the waypoint TDs (K27). The corresponding xy coordinates are calculated using the Daisy Chain (K22) function from the local origin or adjacent waypoint. The offset of the waypoint from the aid-to-navigation is estimated from the navigation chart for the area and used in the Move (K24) function to calculate the waypoint TDs.

The TD data collected in the area around the waypoint is converted to xy and along/cross track position coordinates using function K17, Convert TD Data to xy and AT/CT. The converted data is plotted and compared to the navigation chart. Function K18, Plot XY Data, will plot the position data on the CRT or 9872A plotter. The plot can be scaled to a navigation chart scale (e.g., 1:10,000, 1:20,000, 1:40,000, 1:80,000) or automatically scaled for the range of xy data. The function K19, Plot Along/Cross Track Data, automatically scales and plots the calculated cross track vs along track data.

The xy and along track plots are compared to the navigation chart (and COE dredging charts if available). If the plotted data is consistent with the chart, then the calculated waypoint TD coordinates are good. If the plotted data and navigation chart do not agree, the offset of the waypoint is determined from the plots. The waypoint is "moved" using K24 and the xy and cross track data is recalculated and replotted. This process is repeated until the plotted data greees with the navigation chart.

•Store the TDs of a fixed or floating aid-to-navigation near the waypoint as a temporary waypoint (K23).

•Offset the TDs from the fixed or floating aid near the waypoint to the waypoint (K24).

•Convert TD data collected along channel edges, channel centerlines, and at aids to navigation to xy and along/cross track coordinates (K17).

•If xy and along/cross track data plots (K18, K19) do not agree with "reality," then use MOVE function (K24) to reposition waypoint until data and "reality" agree (repeat K17, K18, K19).

•Store waypoint TD and xy in waypoint file.

WAYPOINT CALCULATIONS - CHANNEL EDGE DATA

Figure 2

WAYPOINT VERIFICATION. There are two general approaches to waypoint verification. If the waypoint was calculated using the channel edge technique: the waypoint calculations are verified in the process of calculating the waypoint. The waypoint TDs are modified until position data converted from TDs for channel edges and aids-to-navigation agreed with the navigation chart.

In the case where the waypoint is defined by the intersection of two ranges, data is collected along the entire length of the channel centerline during the survey. Data is usually also collected along channel edges and at several aids-to-navigation. Data collected along the channel centerline is converted to xy and along/cross track coordinates using function K17. The plot of cross track vs along track (K19) is the best indicator of waypoint quality. If both waypoints are good and the vessel track was close to the centerline over the entire length of the channel, the cross track distance will be zero or near zero at both endpoints (e.g., at the waypoints). There may be a slight "bow" in cross track near the midpoint of the channel length. If the bow is severe a trackpoint can be inserted between waypoints. If the cross track distance is not near zero at one or both waypoints, recheck the waypoint calculations and check the survey notebook for comments as to whether the vessel was on range. It is good survey practice to collect data along the entire length of the channel several times during the survey. Several of these runs should be checked for consistent results before it is concluded that a waypoint is in error or a trackpoint is necessary. Data along channel edges and at aids-to-navigation is also converted to xy and along/cross track position and plotted (K18 and K19). The plots are compared to the navigation charts and COE data for agreement.

Another check of waypoint calculations is to calculate xy positions (using 3 TDs) for TD data collected in an area near the waypoint and use the project function, K27, to calculate the average TD fix triangle. The project function is the inverse of the Convert TD to XY and AT/CT function (K17). For each xy position, a set of TD values is calculated based on the distance from the waypoint position and the waypoint TDs. These calculated TDs are differenced from the measured TDs for each data sample. Statistics for the resultant differences are calculated and a summary table is printed on the hard copy printer.

TRACKPOINT CALCULATION. The location of a trackpoint is determined from the along/cross track plot (K19). The trackpoint is generally selected where the "bow" in the plot is the maximum. The Move function, K24, in the range bearing mode is first used to calculate the "ideal" TD and xy at the range from the waypoint where the trackpoint is to be located. The changes in TD and xy calculated by K24 are applied to the waypoint values and stored in the waypoint table, K27, as the TD and xy for the trackpoint. These trackpoint TD and xy coordinate values are an interpolation between the two waypoints bracketing the channel. To provide a correction for the "bow" in the along/cross track plot, the Move function is used to offset the trackpoint in the direction and equal to the magnitude of the "bow." The resultant calculations are applied to the trackpoint. The trackpoint coordinates are then used as the "to" waypoint for Td to xy calculations and the resultant along/cross positions are replotted as a check of the trackpoint calculations. In some cases there may be a requirement for more than one trackpoint. Additional trackpoints are calculated as outlined above using the first trackpoint as one of the two waypoints.

Rows 1-20 of the waypoint table are generally set aside for waypoints, row 25 contains the data for the local origin. Rows 21-24 are used for trackpoints.

TRACK POINT WORKSHEET

Track point calculations:

- Trackpoint between WP ______ and WP ______
- Distance of Track point from WP ______, R= ______
- Bearing from WP ______, B=
- Results of move of R&B from WP(   ):

	dTDX =	dTDY =	dTDZ =
WP( )	TDX =	TDY =	TDZ =
First Estimate of Trackpoint	=	,	,

WP( ) Position	X =	Y =
	dx =	dy =
First Estimate of Trackpoint XY	=	,

- Trackpoint stored as WP ______
- Magnitude of offset (e.g., BOW), M = ______
- Move Trackpoint; R = M, B = B+90
- Recalculate xy data using trackpoint
- Replot CT/AT data
- Reminder: Re-store waypoint table on waypoint file

Data Analysis Software

Function: Read Data File

Special function key: KO

Subprograms:

```
Read (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
Read_alt(W(*), X(*), Y(*), Z(*), U$(*), N, F$)
```

The program function reads time-difference (TD) and time-of-day data stored on magnetic tape. The TD data may be corrected for known errors. If TDs are corrected, a lower case "t" is annotated to F$.

Input parameters:

```
F$        - File name
Td_cor$   - Indicates if TD data is to be corrected, "Y" = yes, "N" = no.
Wcor      - correction to W(*)
Xcor      - correction to X(*)
Ycor      - correction to Y(*)
Zcor      - correction to Z(*)
```

Output parameters:

```
W(*)  - TDW data
X(*)  - TDX data
Y(*)  - TDY data
Z(*)  - TDZ data
R1(*) - Range 1 data (Not used)
R2(*) - Range 2 data (Not used)
U$(*) - Time-of-day data
N     - number of samples
F$    - annotated file name.
     F$&"t"  = TD data corrected
```

User Instructions:

Prerequisite functions: None

1. Insert data tape into left hand tape transport.

2. Press KO. The menu is cleared and "READ DATA FILE" appears on the CRT in inverse video.

3. When "DATA FILE TYPE? TD&RANGE1 OR TD ONLY2" appears on the display line:

   a. If the data file contains both TD and range data arrays (e.g. post Delaware data or reformatted Delaware data):

(1) Enter: 1
(2) Press: CONT

b. If the data file contains only TD arrays (e.g. Delaware and pre-Delaware data)
   (1) Enter: 2
   (2) Press: CONT

4. When "File Name?" appears in display area:

   a. Enter: File name
   b. Press: CONT

The file is read and the file name and number of samples are printed.

5. When "CORRECTION TO TDs? Y/N" appears in display area:

   a. If you want to correct TD data:

      (1) Enter: Y
      (2) Press: CONT
      (3) Go to Step 6

   b. If you do not want to correct TD data:

      (1) Enter: N
      (2) Press: CONT
      (3) Go to Step 10

6. The start and stop times for the data collected will be printed. When "CORRECTION TO TDW(MICROSEC)?" appears in display area:

   a. Enter: correction to TDW data in microseconds
   b. Press: CONT

7. When "CORRECTION TO TDX(MICROSEC)?" appears in display area:

   a. Enter: correction to TDX data in microseconds
   b. Press: CONT

8. When "CORRECTION TO TDY(MICROSEC)?" appears in display area:

   a. Enter: correction to TDY data in microseconds
   b. Press: CONT

9. When "CORRECTION TO TDZ(MICROSEC)?" appears in display area:

   a. Enter: correction to TDZ data in microseconds
   b. Press: CONT

The corrections entered for each TD will be printed and titled.

10. The program function is complete. The function menu is printed on the CRT.

Function: Calculate Statistics and Linear Regression of TD Data

Special Function Key: K2

Subprograms:

    Hi_lo (W(*), N, Bw, Lw)
    Stat_mat W(*), X(*), Y(*), Z(*), Stat(*), Cov(*), O_set(*), N)
    Reg (Stat(*), Cov(*), S(*), R(*), O_set(*))
    Print (Stat(*), O_set(*), Cov(*), R(*), N, V)

This program function calculates statistics, linear regression coefficients, and minimum and maximum for the four TD arrays. Statistics and regression parameters calculated are:

a. mean for each TD
b. standard deviation for each TD
c. covariance of TD pairs
d. correlation coefficient of each TD pair
e. linear regression slopes of each TD pair
f. standard deviation of residuals for each regression line

Input parameters:

W(*), X(*), Y(*), Z(*)	- TD arrays
N	- number of samples

Output parameters:

Stat(*)	- Summary statistics array containing sums, sums of squares, mean and standard deviation
O_set(*)	- First data sample for each TD
Cov(*)	- Summary statistics array contains sums of cross products, covariance, and correlation coefficients of TD pairs
Bw, Bx, By, Bz	- Maximum value for each TD array
Lw, Lx, Ly, Lz	- minimum value for each TD array.
S(*)	- A summary statistics array containing mean and standard deviation for each TD pair
R(*)	- An array containing linear regression slope, RMS residuals, and definition of the independent variable for each TD pair

<u>User Instructions</u>:

Prerequisite functions: K0 Read Data

1. Press K2. The menu is cleared and "STATS AND REGRESSION OF TD DATA" appears on the CRT in inverse video.

2. The file name, start time, stop time, and a tabulation of the statistics and linear regression parameters for the TDs and TD pairs is printed on the hard copy printer.

3. The program function is complete. The function menu is printed on the CRT.

Function: Separate Data into Subfiles

Special Function Key: K3

Subprograms:

Separate (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*))
Store (A(*), B(*), C(*), D(*), R1(*), R2(*), U$(*), Nt, F$)

This program function enables the operator to separate and store a large data set into several smaller data files or to store a subset of the data into a data file.

Function parameters:

W(*), X(*), Y(*), Z(*) - TD data arrays
R1(*), R2(*) - Miniranger data arrays
U$(*) - Time-of-day data array
A(*), B(*), C(*), D(*) - Temporary TD data arrays

User Instructions:

Prerequisite functions: Read Data, K0, or Link Data, K25

1. Press K3. The menu is cleared and "SEPARATE DATA INTO SUBFILES" appears on the CRT.

2. When "FOR SUBFILE ENTER: START SAMPLE NUMBER" appears on the CRT:

   a. Enter: The sample number for the beginning of the subset of interest
   b. Press: CONT

3. When "FOR SUBFILE ENTER: STOP SAMPLE NUMBER" appears on the CRT:

   a. Enter: The last sample number for the subset of interest
   b. Press: CONT

4. When "FILE NAME?" appears on the CRT:

   a. Enter: File name for subset to be stored. Ensure that a tape cartridge with available space is in the left hand tape drive.
   b. Press: CONT

5. The data file is created and data stored. The menu is reprinted on the CRT. If another subfile is to be stored, go to Step 1.

Function: Plot TD Data with Regression Lines

Special Function Key: K4

Subprograms:

Plot (Lx, Bx, Ly, By, Xaxis, Yaxis, X(*), Y(*), Slope, N, "TDX", "TDY", F$)

The program function plots two of the time-difference arrays against each other and also plots the linear regression line for the TD pair chosen. The program automatically scales the plot for the minimum and maximum TDs for each array. The axes are drawn through the mean for each TD. Minor tic marks are every microsecond and major tic marks every 10 microseconds.

Parameters:

Plot - TD pair to be plotted

1 = WX; Z = WY; 3 = WZ; 4 = XY, 5 = XZ, 6 = YZ

Lw, Lx, Ly, Lz - Minimum TD for each array

Bw, Bx, By, Bz - Maximum TD for each array

S(*) - Statistics summary array containing means and standard deviations for each pair to TD arrays

R(*) - Regression summary array which contains slopes of regression lines

N - Number of samples

F$ - File name

<u>User Instructions:</u>

Prerequisite function: K0 - Read Data File
K2 - Calculate Statistics and Linear Regression of TD Data

1. Press K4. The menu is cleared and "PLOT TD DATA WITH REGRESSION LINES" appears on the CRT in inverse video.

2. When "PLOT? WX1, WY2, WZ3, XY4, XZ5, YZ6" appears on the display line:

   a. Enter: 1-6 depending on which plot is desired
   b. Press: CONT

3. The selected TD data will be plotted on the CRT. When finished viewing the plot:

   a. Press: CONT

Function: Plot Residuals

Special Function Key: K5

Subprograms: Rplot(Iv(*), D(*), S(*), R(*), Pr, N, V1$, V2$, F$)
Hi_lo(X(*), N, Hi, Lo)

This program function plots the residuals from the linear regression of any TD pair (See K2). The residuals may be plotted against sample number or the independent variable. Residuals are normalized to the standard deviation of the residuals. Normalized values greater than 5 are printed on the hard copy printer and are not plotted.

Variables:

W(*), X(*), Y(*), Z(*) - TD data arrays
S(*) - Statistics summary array
R(*) - Regression summary array
Pr - TD pair for which residuals are being plotted: 1 = WX, 2 = WY, 3 = WZ, 4 = XY, 5 = XZ; 6 = YZ
N - Numbers of Samples
V1$, V2$ - Names of two TDs for which residuals are being plotted, i.e. "TDW", "TDX", etc.
F$ - Data file name

User Instructions:

Prerequisite functions:
K0 - Read Data
K2 - Calculate Statistics and Linear Regression of TD Data

1. Press K5. The menu is cleared and "PLOT RESIDUALS" is printed on the CRT in inverse video.

2. When "PLOT? WX1, WY2, WZ3, XY4, XZ5, YZ6" appears on the display line:

   a. Enter: 1-6 depending on which plot is desired
   b. Press: CONT

3. When "PLOT RESIDUALS VS N(1) OR INDEPENDENT VAR(2)?" appears on the display line:

   a. To plot the residuals vs sample number:
      (1) Enter: 1
      (2) Press: CONT
      (3) Go to: Step 4
   b. To plot the residuals vs the independent variable:
      (1) Enter: 2
      (2) Press: CONT
      (3) Go to: Step 4

4. The normalized residuals for the selected TD pair regression line will be plotted on the CRT. When finished viewing the plot:

a. Press: CONT

5. When "HARD COPY? Y OR N" appears on the display line:

   a. If a hard copy of the plot is desired:
      (1) Enter: Y
      (2) Press: CONT
   b. If no hard copy of the plot is desired:
      (1) Enter: N
      (2) Press: CONT

6. The program function is complete. The function menu is printed on the CRT.

Function: Edit Data

Special Function Key: K16

Subprograms:

```
Delete (W(*)), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
Delete_blok (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
Delete_td (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
```

This program function is used to edit (i.e. remove) samples from the data (TD, time, and range) arrays. Three options are available to the user:

a. The first option deletes a single data sample in each data array.
b. The second deletes a block of data samples.
c. The third deletes samples with TD samples outside a range that is input by the operator.

When editing of the data arrays is complete, the operator may store the edited data on a new file for later use.

NOTE: The edit function deletes only data samples. Arrays containing XY and along/cross track position data are not affected. Function K17 must be repeated to see the effect of editing on computed positions.

Paramaters:

Edit - Indicates which edit subroutine is to be used.

1 - delete sample by sample
2 - delete block of samples
3 - delete samples with TD values outside limits determined by operator

W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*) - Data arrays
N - Number of samples
F$ - Data file name

<u>User Instructions</u>:

Prerequisite functions:

K0: Read Data
Optional: Several other program functions may be run prior to the Edit function to determine which data samples are to be edited, e.g. K2, K4, K5, K18

1. Press: K16. The menu is cleared and "EDIT DATA" appears on the CRT in inverse video.

2. When "SINGLE LINE1, BLOCK2, OR TD-CLIP3" appears on the display line:

a. If you want to delete samples one at a time:
   (1) Enter: 1
   (2) Press: CONT
   (3) Go to Step 3
b. If you want to delete a block(s) of samples:
   (1) Enter: 2
   (2) Press: CONT
   (3) Go to Step 8
c. If you want to delete samples outside a set of TD bounds:
   (1) Enter: 3
   (2) Press: CONT
   (3) Go to Step 11

3. When "SAMPLE TO BE DELETED? START WITH HIGHEST NUMBER" appears on the display line:

   a. Enter: Sample number
   b. Press: CONT

The sample deleted and the number of samples remaining are printed on the hard copy printer.

4. When "ANOTHER SAMPLE TO DELETE?" appears in the display area:

   a. Enter: Y(if another sample is to be deleted) or N
   b. Press: CONT
   c. If another sample is to be deleted, go to Step 3.

5. When "LIST DATA?" appears in the display area:

   a. If a listing of the data is desired:
      (1) Enter: Y
      (2) Press: CONT. The data is printed on the hard copy printer.
   b. If a listing of the data is not desired:
      (1) Enter: N
      (2) Press: CONT

6. When "STORE DATA?" appears in the display area:
   a. If you want to store the data
      (1) Enter: Y
      (2) Press: CONT
      (3) Go to Step 7
   b. If you do not want to store the data
      (1) Enter: N
      (2) Press: CONT
      (3) Go to Step 15

7. Insure that a tape cartridge is loaded into the left hand tape drive. When "FILE NAME?" appears on the display area:
   a. Enter: File name
   b. Press: CONT
   c. Go to Step 15

8. When "FIRST SAMPLE IN BLOCK?" appears in the display area (if more than one block is to be deleted, delete the block with the higher sample numbers first. The array is renumbered after each edit.)

a. Enter: First sample
b. Press: CONT

9. When "LAST SAMPLE IN BLOCK TO BE DELETED" appears in the display area:

a. Enter: Last sample
b. Press: CONT

The block of samples deleted and the numbers of samples remaining are printed on the hard copy printer.

10. When "ANOTHER BLOCK TO BE DELETED?" appears in the display area

a. If there is more data to be deleted
   (1) Enter: Y
   (2) Press CONT
   (3) Go to Step 8
b. If no more data is to be deleted
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 5

11. When "CLIP LIMITS FOR TDW? MIN, MAX" appears in the display area:
a. Enter: minimum TDW, maximum TDW
b. Press: CONT

12. When "CLIP LIMITS FOR TDX? MIN, MAX" appears in the display area:
a. Enter: minimum TDX, maximum TDX
b. Press: CONT

13. When "CLIP LIMITS FOR TDY? MIN, MAX" appears in the display area:
a. Enter: minimum TDY, Maximum TDY
b. Press: CONT

14. When "CLIP LIMITS FOR TDZ? MIN, MAX" appears in the display area:
a. Enter: minimum TDZ, maximum TDZ
b. Press: CONT

The number of samples deleted and number of samples remaining are printed on the hard copy printer.
c. Go to Step 5

15. The program function is complete. The program menu is printed on the CRT.

Function: Convert TD Data to XY Position

Special Function Key: K17

Subprograms:

```
Cart_coord (Xmit(*), Wpt(25,7), Wpt(25,8), Zxmit(*))
Rb (Wpt(Wt,5), Wpt(Wt,6), Zmit(I,1), Zxmit(I,2),Bear(I), Range (I))
G_mat (Power(*), V, Range(*), Bear(*), Zp(*), Aa(*), G123(*), G12(*),
  G23(*), G13(*))
Wp3 (Wpt(*), Conf, Wt, Tp(*), Zpp(*), L$)
Td3 (W(*), X(*), Y(*), Z(*), I, Conf, Tq(*))
Fehg (Zxmit(*), Zpp(*), Tp(*), Tq(*), G123(*), Zq(*), V)
Fehgt (Zxm(*), Zpp(*), Tp(*), Tq(*), G(*), Zq(*), V)
Wp2 (Wpt(*), Conf, Wt, Tpp(*), Pair, Zpp(*), G12(*), G23(*), G13(*), G(*),
  L$, Zxmit(*), Zxm(*))
Td2 (W(*), X(*), Y(*), Z(*), I, Conf, Pair, Tqq(*))
Wpf2 (Wpt(*), Conf, Pair, Wf, Tqq(*))
Ct_at (Pos_x(*), Pox_y(*), Ct(*), At(*), Wpt(Wt,5), Wpt(Wt,6)
Stat_mat (Pos_x(*), Pos_y(*), Ct(*), At(*), Stat(*), Cov(*), O_set(*), N)
Reg (Stat(*), Cov(*), S(*), R(*), O_set(*))
Track (Stat(*), O_set(*), Cov(*), R(*), Sample, F$, Wt, Wf, Angle, N)
Rotate (Glat, Glon, R(*))
Cart (Plat, Plon, X, Y)
FNGlat (Lat, F)
Pseudo (Glat, Glon, R*, Plat, Plon)
```

This program function converts Td data to xy and along/cross track positions using the FEHG algorithm and surveyed waypoints. The program will compute either a three or two TD solution. A summary table is printed on the hard copy printer which lists:

- -chain and LOPs used in the solution
- -file name
- -bearing angle between waypoints used for along/cross track calculation
- - rms trackline of data
- -average cross track position
- -standard deviation of cross track position
- -average xy position
- -standard deviation of xy position data

Function Parameters:

Chain$ - Loran-C transmitting chain file which contains transmitter positions, power, and emission delays. The convention for naming chain files is a four letter code followed by a number from 1 to 4, eg NEUS1. The number designates the three secondaries utilized: 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ.
C1 - Flag which is set when the Loran-C chain file is read
Pos_x(*), Pos_y(*) - x and y position arrays
At(*), Ct(*) - Along and cross track position arrays
Xmit(*) - Transmitter geodetic positions
Power(*) - Transmitter power levels
Emis(*) - Secondary transmitter emission delays
F1 - Flag to indicate when the waypoint file is read
File$ - Waypoint file name

Wpt(*) - Waypoint data array
Zxmit(*) - Transmitter xy positions referenced to local origin (Wpt(25,7), Wpt(25,8))
Wt - Waypoint used as reference for TD to Xy calculations
Wf - Waypoint used for along/cross track calculations to Wt
Bear(*) - Bearings from waypoint to transmitters
Range(*) - Ranges from waypoint to transmitters
Zp(*) - XY position of waypoint
V - Velocity of propagation
Aa(*) - Gradient matrix, xy to TD
G123(*) - Three Td gradient matrix, TD to XY
G12(*), G13(*), G23(*) - Two TD gradient matrices, TD to XY
Conf - Chain configuration 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ
Ch$ - Abbreviation of chain, eg first four characters of chain$
So - Indicates two or three solution: 2 = 2-TD, 3 = 3-TD
L$ - LOPs used in TD to XY calculation, eg XYZ, YZ, WX, etc.
W(*), X(*), Y(*), Z(*) - TD data arrays
Zq(*) - XY position solution
Tq(*) - 3-TD sample
TQQ(*) - 2-TD sample
Zpp(*) - XY position of waypoint
Tp(*) - 3-TD waypoint
Tpp(*) - 2-TD waypoint
Zxx(*) - Dummy position array
Pair - LOP pair used for TD to xy calculation
  If Conf = 1; 1 - XY, 2 = XZ, 3 = YZ
  If Conf = 2; 1 = WX, 2 = WY, 3 = XY
  If Conf = 3; 1 = WX, 2 = WZ, 3 = YZ
  If Conf = 4; 1 = WY, 2 = WZ, 3 = YZ
G(*) - 2TD G matrix used in 2TD solution
Zxm(*) - Transmitter xy positions - two TD case
R - Distance between "TO" and "FROM" waypoints
Stat(*), Cov(*), O_set(*), S(*), R(*) - Arrays containing summary statistics of xy and along/cross track position

<u>User Instruction:</u>

Prerequisite instructions: Read Data, K0

1. Press K17. The menu will be cleared and "CONVERT TD TO XY" will appear on the CRT in inverse video.

2. When "CHAIN CONFIGURATION? " appears on the display line:

(1) Insure that the tape cartridge containing chain data is in the right hand tape drive.
(2) Enter: Chain file. Note: Chain files have five character names. The first four characters denote the chain, eg NEUS, SEUS, WCUS. The number following the characters denotes the secondaries, eg: 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ
(3) Press: CONT

NOTE: Chain configuration data is only read once. The data is stored in memory for further use. To change chain configuration either:

a. Press: STOP
b. Press: RUN. This clears data memory and "CHAIN CONFIGURATION" will be requested as above. All other data in memory will also be cleared.

OR

a. Enter: C1 = 0
b. Press: EXECUTE. This clears the flag, C1, which will cause the program to branch through the "CHAIN CONFIGURATION?" statement.

3. When "WAYPOINT FILE NAME?" appears in the display area:

   a. Insure that the tape cartridge containing waypoint data is in the right hand tape drive.
   b. Enter: Waypoint file
   c. Press: CONT

NOTE: Waypoint file data is read only once. The data is stored in memory for further use.

To change the waypoint file:

a. Press: STOP
b. Press: RUN. This clears data memory and "WAYPOINT FILE" will be requested as above

OR

a. Enter: F1 = 0
b. Press: EXECUTE. This clears the flag F1 which will cause the program to branch through the "WAYPOINT FILE?" statement.

4. When "WAYPOINT TO?" appears in the display area:

   a. Enter: Waypoint number. NOTE: If a comparison of Mini-ranger and Loran-C position data is to be made, insure that "TO" and "FROM" waypoints are consistent. Conversion waypoint is "TO" waypoint.
   b. Press: CONT

5. When "WAYPOINT FROM ?" appears in the display area:

   a. Enter: Waypoint number. See Note above
   b. Press: CONT

6. When "TWO OR THREE TD SOLUTION?" appears in the display area:

   a. If a two TD solution is desired:
      (1) Enter: 2
      (2) Press: CONT
      (3) "TWO TD SOLUTION, CHAIN =   LOPS =   " is printed on the hard copy printer.
   b. If a three TD solution is desired:
      (1) Enter: 3
      (2) Press: CONT
      (3) "THREE TD SOLUTION, CHAIN =   LOPS =   " is printed on the hard copy printer.

7. The sample number is displayed in display area as the program sequences through the data arrays. When the calculations are complete, a summary table is printed on the hard copy printer which contains:

a. data file name
b. bearing angle between waypoints
c. rms trackline of data
d. average cross track position
e. standard deviation of cross track position
f. average x,y position
g. standard deviation of xy position data

8. The program function is complete. The function menu is printed on the CRT.

Function: Plot XY Data

Special Function Key: K18

Subprograms:

Hi_lo (Pos_x(*), N, Maxx, Minx)
Xplot (T, Pos_x(*), Pos_y(*), Min_x, Max_x, Min_y, Max, y, N, No, F$, Ch$, L$, Pl)

This program function plots the xy data calculated from Loran-C TDs on the CRT or 9872A plotter. Two options are available. The first automatically scales the CRT (or9872A) plotting area to the range of xy data. The operator may zoom in on a section of the plot, find the sample number of a plotted point, and digitize up 10 locations on the plot. The second option plots the xy data to a chart scale (1:10,000,1:20,000, 1:40,000, or 1:80,000). The axes are drawn through a waypoint selected by the operator. The operator also inputs the offset of the waypoint (axes) from the lower left hand corner of the plotting area. The selection of waypoint offset and scale determines the window of data which will be plotted.

Function parameters:

Pos_x(*), Pos_Y(*) - xy position calculated from Loran-C data
Minx, Miny - Minimum x and y positions
Maxx, Maxy - Maximum x and y positions
T - Line type
N - Number of samples
No - Denotes first or second time data is being plotted
Ch$ - Loran-C chain, eg NEUS, SEUS, etc.
L$ - Loran-C LOPs, eg XY, WX, etc.

User Instructions:

Prerequisite functions: K0 and K17

1. Press K18: The menu is cleared and "PLOT XY DATA" is printed on the CRT.

2. When "STANDARD XY PLOT1 OR PLOT TO CHART SCALE2? 1 or 2" appears on display line:

   a. If data to be plotted standard X Y
      1. Enter 1
      2. Press: CONT
      3. Go to step 3
   b. If data to be plotted to chart scale
      1. Enter: 2
      2. Press: CONt
      3. Go to step 15

3. When "PLOT DATA ON CRT1 OR 9872A2: 1 OR 2" appears on the display line:

a. Enter: 1 or 2
b. Press: CONT

4. After viewing plot

a. Press: CONT

5. When "ZOOM: Y OR N" appears on display line:

NOTE: When using 9872A, change paper

a. To blow up a section of the plot:
   (1) Enter: Y
   (2) Press: CONT
   (3) Go to step 6 or 7
b. If a "zoom" or further "zoom" is not wanted:
   (1) Enter: N
   (2) Press: CONT
   (3) Go to step 8

6. For CRT, when the cursor appears on the CRT, place cursor at the lower left-hand corner of the area of interest and press CONT. When cursor reappears, position cursor at the upper right-hand corner of the area of interest and press CONT. The selected area is replotted on the CRT.

Go to step 5

7. For 9872A, place pen at lower left-hand corner of the area of interest and press"ENTER" on the plotter.

Go to Step 5

NOTE: The "zoom" may be repeated several times.

8. When "FIND SAMPLE NUMBER AND VALUE OF PLOTTED DATA POINT?" appears on the display line:

a. If the sample number of a point on the plot is desired:
   (1) Enter: Y
   (2) Press: CONT
b. If not:
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 11

9. For CRT, when the cursor appears on the CRT:

a. Center it on the point of interest
b. Press: CONT

10. For 9872A, move pen to point of interest

a. Press "ENTER" on plotter

The sample number and coordinates will be printed on the printer. If no data is printed, try repositioning the cursor closer to the point of interest or replot the data and use "zoom" to blow up the area around the point of interest.

c. Go to Step 8

11. When "DIGITIZE?" appears in the display area:

a. If you want to digitize 1-10 points on the plot:
   (1) Enter: Y
   (2) Press: CONT
b. If not:
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 9

12. When "HOW MANY? 1-10" appears in the display area:

a. Enter: the number of points to be digitized
b. Press: CONT

For CRT, when the cursor appears on the CRT, center it over a point of interest and

c. Press: CONT

For 9872A, center pen over point of interest

d. Press: Enter

A plus sign (+) will appear over the data point and the number of points digitized (1-10) will appear to its right. Continue the procedure until the number of points to be digitized is complete. A list of the digitized point coordinates is printed on the hard copy printer.

d. Go to Step 14

13. The plot will reappear on the CRT. When finished viewing the plot press CONT.

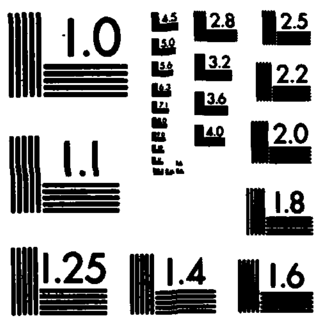
14. For CRT, when "HARDCOPY? Y OR N" appears on the display line:

a. If a hard copy of the plot is desired:
   (1) Enter: Y
   (2) Press: CONT
   (3) Go to step 20
b. If not:
   (1) Enter: N
   (2) Press: CONT
   (3) Go to step 20

AD-A124 343 HHE/LORAN-C SURVEYING(U) COAST GUARD WASHINGTON DC OFFICE OF RESEARCH AND DEVELOPMENT A J SEDLOCK NOV 82 USCG-D-54-82

2/4

UNCLASSIFIED F/G 17/7 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

15. When "WAYPOINT FOR AXES" appears on display line
  a. Enter: Waypoint No. for Axes
  b. Press: CONT

16. When "WAYPOINT OFFSET FROM LOWER RIGHT HAND CORNER OF PLOT? X, Y (KM)" appears on display line:

  a. Enter: X,Y
  b. Press: CONT

17. When "PLOT DATA ON CRT1 OR 9872A2? 1 OR 2" appears on the display line:

  a. Enter: 1 or 2
  b. Press: CONT

18. When "CHART SCALE 1:10,0001, 1:20,0002, 1:40,0004, 1:80,0008, 1, 2, 4, or 8" appears on display line:

  a. Enter: Chart scale (1, 2, 4 or 8)
  b. Press: CONT

19. If data plotted on CRT:

  a. Go to step 20

20. The function is complete. The function menu is printed on the CRT.

Function: Plot Along/Cross Track Data

Special Function Key: K19

Subprograms:

Hi_lo (Ct(*), N, Maxc, Minc)
Patct (At(*), Ct(*), N, Mina, Maxa, Minc, Maxc, Wt, Wf, F$, R, No, Data)
G_mat (P(*), V, Range(*), Bear(*), Zp, A(*), G123(*), G12(*), G23(*), G13(*))

This program function plots cross track vs along track position. The plots are automatically scaled and labeled.

Function parameters:

At(*), Ct(*) - Along and cross track position arrays calculated from Loran-C data
N - Numbers of samples
Maxc, Maxcc - Maximum cross track distance
Minc, Mincc - Minimum cross track distance
Maxa, Maxaa - Maximum along track distance
Mina, Minaa - Minimum along track distance
Wt - Waypoint to
Wf - Waypoint from
R - Distance between waypoints
F$ - File name

User Instructions:

Prerequisite function: K17

1. Press K19. The program menu is cleared and "PLOT ALONG TRACK CROSS TRACK DATA" is printed on the CRT in inverse video.

2. The along/cross track data is plotted on the CRT. When finished viewing the plot:

   a. Press: CONT

3. When "HARD COPY? Y OR N" appears in the display area:

   a. If a hard copy of the plot is desired:
      (1) Enter: Y
      (2) Press: CONT
   b. If not:
      (1) Enter: N
      (2) Press: CONT

4. This function is complete. The function menu is printed on the CRT.

Function: Predict TD

Special Function Key: K20

Subprograms:

Range (Lat, Lon, Xmit(I,1), Xmit(I,2), Range (I))
Bear (Lat, Lon, Xmit(I,1), Xmit(I,2), Bear(I))
Td (Range(*), V, Emis(*), Td(*))
G_mat(P(*), V, Range(*), Bear(*), Zp(*), A(*), G123(*), G12(*), G23(*), G13(*))

This program function calculates predicted TDs for an input latitude, longitude. The program also outputs range and bearing to transmitter stations and GDOPs for the three TD and two TD fixes.

Function Variables:

C1 - Chain flag indicates if chain data has been read into memory
Chain$ - Chain name
Xmit(*) - Transmitter positions (latitude, longitude)
Power(*) - Transmitter power
Emis(*) - Secondary emission delays
P$ - Indicates input position is a waypoint
F1 - Waypoint table flag, indicates waypoint table has been read into memory
Wpt(*) - Waypoint table
Lat,Lon - Latitude and Longitude of position of interest
D,M,S - Degrees, Minutes, Seconds
Range (*) - Ranges to transmitters
Bear(*) - Bearings to transmitters
Td(*) - Predicted TDs
Aa(*) - Gradient matrix; xy to TD
G123(*) - Three TD gradient matrix; TD to XY
G12(*), G13(*), G23(*) - Two TD gradient matrices, TD to XY
Gdop 123 - Three TD Geometric Dilutin of Precision (GDOP)
Gdop12, Gdop13, Gdop23 - Two TD GDOPs
Ch$ - Chain abbreviation, eq NEUS = North East United States
Conf - Indicates three TD configuration: 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ

User Instructions:

Prerequisite functions: None

1. Press K20. The program menu is cleared and "TD PREDICTION" is printed on the CRT in inverse video.

2. When "CHAIN FILE" appears in the display area:

    (1) Insure that the tape cartridge containing chain data is in the right hand tape drive
    (2) Enter: Chain file. Note: Chain files have five character names. The first four characters denote the chain, eg NEUS SEUS, etc. The number following the character denotes the secondaries, eg 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ
    (3) Press: CONT

Note: Chain configuration data is read only once. The data is stored in memory for further use. To change chain configuration either:

a. Press: STOP
b. Press: RUN. This clears data memory and "CHAIN FILE" will be requested as above. All other data in memory will also be cleared

OR

a. Enter: C1 = 0
b. Press: EXECUTE. This clears the flag C1 which will cause the program to branch through the "CHAIN FILE" statement.

3. When "IS POINT OF INTEREST ON WAYPOINT FILE? Y OR N" appears in the display area:

a. If the point of interest is on the waypoint file:
   (1) Enter: Y
   (2) Press: CONT
b. If not:
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 6

4. When "WP FILE NAME?" appears on the display line (see Note below)

a. Insure that the tape cartridge containing waypoint data is in the right hand tape drive
b. Enter: Waypoint file name
c. Press: CONT

Note: This step is skipped on successive runs of this program function or if the waypoint file has been read during a previous function.

5. When "WAYPOINT NO.?" appears on the display line:

a. Enter: the waypoint of interest
b. Press: CONT
c. Go to Step 8

6. When "INPUT LAT IN DEGREES, MIN, SEC" appears on the display line:

a. Enter: Latitude of position of interest (Degrees, Minutes, Seconds). The program assumes North latitude
b. Press: CONT

7. When "INPUT LON IN DEGREES, MIN, SEC" appears on the display line:

a. Enter: Longitude of position of interest (Degrees, Minutes, Seconds). The program assumes East longitude
b. Press: CONT

8. The program calculates and prints the following parameters:

   a. Predicted time differences. (If a waypoint is selected, surveyed TDs are also listed.)
   b. Ranges and bearings to transmitters
   c. GDOPs for the three TD and each of the two TD combinations.

9. The program function is complete. The program menu is printed on the CRT.

Function: Calculate Waypoint

Special Function Key: K21

Subprograms:

```
Read(W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N1, F$)
Read_alt(W(*), X(*), Y(*), Z(*), U$(*), N1, F$)
Stat_mat(W(*), X(*), Y(*), Z(*), Stat(*), Cov(*), O_set(*), N1)
Reg(Stat(*), Cov(*), S1(*), Rone(*), O_set(*))
Print(Stat(*), O_set(*), Cov(*), Rone(*), N1, 1)
Way(S1(*), S2(*), Rone(*), Rtwo(*), N1, N2, Wp(*))
Wprint(Wp(*))
```

This program function computes estimated waypoint time differences, estimated rms error, and lop crossing angles. The function also calculates and prints statistics and regression parameters for each of the survey tracklines.

Parameters:

Read - Type of data file

W(*), X(*), Y(*), Z(*) - TD data arrays
R1(*), R2(*) - Range data arrays
U$(*) - Time array
N1 - Number of samples for first data file
N2 - Number of samples for second data file
Stat(*), Cov(*), O_set(*) - Statistics arrays
S1(*), Rone(*) - Statistics summary arrays for first data file
S2(*), Rtwo(*) - Statistics summary arrays for second data file
Wp(*) - Summary of waypoint calculations

User Instructions:

Prerequisite functions: None required

1. Press K21. The menu is cleared and "CALCULATE WAYPOINT" is printed on the CRT.

2. When "DATA FILE TYPE? TD&RANGE 1 OR TD ONLY 2 ?" appears on the display line:

   a. If the trackline data files contain both TD and range arrays (i.e. post Delaware data or reformatted Delaware data)
      (1) Enter: 1
      (2) Press: CONT

   b. If the trackline data files contain only TD arrays (e.g. Delaware River data):
      (1) Enter: 2
      (2) Press: CONT

3. When "FILE NAME" appears on the display line:

a. Enter: file name of one of the survey tracklines
b. Press: CONT

The file name is printed on the hard copy printer. Statistics and regression parameters are calculated for the file data and the results are printed on the hard copy printer.

4. When "FILE NAME" reappears on the display line:

a. Enter: File name of the other survey trackline
b. Press: CONT

The file name is printed on the hard copy printer. Statistics and regression parameters are calculated for the file data and the results are printed on the hard copy printer.

5. The intersection of each pair of regression lines is calculated along with the crossing angles and estimated rms error of each solution. The results are printed in tabular form on the hard copy printer.

6. The function is complete.

Function: Daisy Chain

Special Function Key: K22

Subprograms:

Cart_coord (Xmit(*), Wpt(25,7), Wpt(25,8), Zxmit(*))
Rb(Wpt(Rw,5), Wpt(Rw,6), Zxmit(I,1), Zxmit(I,2), Bear (I), Range (I))
G_mat (Power(*), V, Range(*), Bear(*), Zp(*), Aa(*)), G123(*), G12(*), G23(*), G13(*))
Wp2 (Wpt(*), Conf, Rw, Tpp(*) Pair, Zpp(*), G12(*), G23(*), G13(*), G(*), L$,Zxmit(*), Zxm(*))
Wpf2 (Wpt(*), Conf, Pair, Iw, Tqq(*))
Fehgt (Zxm(*), Zpp(*), Tpp(*), Tqq(*), G(*), Zq(*), V)
Wp3 (Wpt(*), Conf, Rw, Tp(*), Zpp(*), L$)
Fehg (Zxmit(*), Zpp(*), Tp(*), Tq(*), G123(*), Zq(*), V)
Rotate (Glat, Glon, R(*))
Cart (Plat, Plon, X, Y)
FNGlat (Lat, F)
Pseudo (Glat, Glon, R*, Plat, Plon)

This program function calculates the position of a waypoint based on the difference in TDs between it and a neighboring waypoint. The Flat Earth Hyperbolic Grid (FEHG) is used to calculate position coordinates. Differential xy coordinates, differential latitude and longitude, and range and bearing between waypoints is also calculated. The user has the option of inserting the calculated xy coordinates (and latitude longitude) into the waypoint table, Wpt(*). Note: This function does not restore the new waypoint table in magnetic tape. If this is desired, function K23 must be used.

Function variables:

C1 - Status variable which indicates if chain data has been read into memory
F1 - Status variable which indicates if the waypoint table has been read into memory
Chain$ - Chain file
Xmit(*), Power(*), Emis(*) - Chain variables; geodetic coordinates, power levels, and emission delays
Conf - Chain configuration
Ch$ - First four characters of Chain$
Wpt(*) - Waypoint table
Zxmit(*) - Transmitter xy coordinates referenced to local origin (three TD case)
Rw - Reference waypoint (start)
Iw - Waypoint of interest (stop)
Bear(*), Range(*) - Bearings and ranges to transmitters
Zpp(*), Zp(*) - Reference waypoint position
V - Velocity of propagation
Aa(*), G123(*), G12(*), G13(*), G23(*) - gradient matrices
Zxm(*) - Transmitter xy coordinates referenced to local origin (two TD case)
So - Indicates two or three TD solution of FEHG
Tpp(*), Tp(*) - Two and three TD waypoints (start)
Tqq(*), Tq(*) - Two and three TD waypoints (stop)
Pair - TD pair (two TD case)

G(*) - Gradient matrix associated with TD pair (Pair)
L$ - Lops e.q. "XYZ," "XY," "XZ," etc.
Zq(*) - Waypoint of interest x, y position
Zxx(*) - Dummy position array
Dx, Dy - Difference in xy position of waypoints
Dlat, Dlon - Difference in latitude, longitude of waypoints
Lat, Lon - Computed latitude, longitude of waypoint
A - Angle between waypoints
R - Range between wayopints

User Instructions:

Prerequisite functions: assumes a waypoint file has been created and it contains TDs for the two waypoints and an xy positions for the reference waypoint.

1. Press K22. The program menu is cleared and "DAISY CHAIN" appears on the CRT in inverse video.

2. If "CHAIN FILE?" appears in the display line:

   (1) Enter: Five character chain file (e.g. NEUS2, GTLK1, etc.) Insure that the tape cartridge containing the file is in the right hand tape drive.
   (2) Press: CONT

3. If "WP FILE?" appears in the display area:

   a. Enter: Waypoint file name. Insure that the tape cartridge containing the waypoint file is in the right hand tape drive.
   Press: CONT

4. When "INPUT START WAYPOINT NO." appears in the display area:

   a. Enter: Start or reference waypoint
   b. Press: CONT

5. When "END WAYPOINT" appears in the display area:

   a. Enter: Waypoint of interest
   b. Press: CONT

6. When "TWO OR THREE TD SOLUTION, 2 OR 3" appears in the display area:

   a. If a three TD solution is desired:
      (1) Enter: 3
      (2) Press: CONT
      (3) Go to Step 8

b. If a two TD solution is desired
   (1) Enter: 2
   (2) Press: CONT

7. When "INPUT TD PAIR; XY1, XZ2, XY3" appears in the display area: (The choice of TD pairs depends on the chain configuration, e.g. if Conf = 2, the choices are WX, WY, XY)

   a. Enter: Number corresponding to TD pair desired
   b. Press: CONT

8. A table is printed on the hard copy printer which contains the following information:

   a. the chain and lops used to calculate the waypoint position
   b. the number of the waypoint used as the origin of the local grid
   c. the two waypoints numbers
   d. the differential xy positions and differential latitude, longitude
   e. the angle and range between waypoints
   f. the projected waypoint position coordinates based on the waypoint TDs
   g. the current values of the waypoint position coordinates

9. When "CHANGE POSITION AND LAT/LON OR WAYPOINT OF INTEREST? Y OR N" appears on the display line:

   a. If you want to change the values
      (1) Enter: Y
      (2) Press: CONT. Note: the values will be changes in the waypoint table, but not on magnetic tape. Function K23 must be used to restore the table on tape.
      (3) "WAYPOINT TABLE CHANGED" is printed on the hard copy printer.

10. The program function is complete. The CRT is cleared and the program menu is printed.

Function: TD MOVE

Special Function Key: K24

Subprograms:

Cart_coord (Xmit(*), Wpt(25,7), Wpt(25,8), Zxmit(*))
Rb (Zp(1), Zp(2), Zxmit(I,1) Zxmit(I,2), Bear(I), Range(I))
Rotate (Glat, Glon, R(*))
Cart (Plat, Plon, X, Y)
FNGlat (Lat, F)
Pseudo (Glat, Glon, R(*))

This program function calculates the change in TD from a waypoint to a position offset from the waypoint. This offset in waypoint position and TD may be applied to the waypoint table. The change in TDs is calculated based on the change in distances to the transmitters.

Function Variables:

C1 - Status variable which indicates if chain data has been read into memory
F1 - Status variable which indicates if the waypoint table has been read into memory
Chain$ - Chain file, e.g. NEUS1, GTLK2, etc.
Ch$ - First four characters of Chain$
Conf - Chain configuration, fifth character of Chain$, 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ
Xmit(*), Power(*), Emis(*) - Transmitter data: geodetic position, power level, emission delay
Wpt(*) - Waypoint table
Wp - Waypoint of interest
Move - Indicates if change in position is to be input as dx, dy or range/bearing
Dx, Dy - Offset in x and y direction from waypoint
R, B - Range and bearing of offset position from waypoint
Zp(*) - Waypoint position and offset position
Zxmit(*) - xy coordinates of transmitters relative to local origin
Bear(*), Range(*) - Ranges and bearings to transmitters
T(*) - Variable to store intermediate calculations and TD offsets
File$ - Waypoint file name
Cor$ - Input variable which indicates if the offset is to be applied as a correction to the waypoint TD in the waypoint table

<u>User Instructions</u>:

Prerequisite function: none. It is assumed that a waypoint file exists.

1. Press K24. The program menu is cleared and "MOVE: CALCULATES <u>Td</u>q = <u>Td</u>p+h(<u>Z</u>q)-h(<u>Z</u>p)" appears on the CRT.

2. If "CHAIN FILE?" appears on the display line:

   (1) Insure tape containing chain file is in right hand tape drive
   (2) Enter: Chain file (e.g. NEUS2, GTLK1)
   (3) Press: CONT

3. If "WAYPOINT FILE?" appears on the display line:

   a. Insure the tape cartridge containing the waypoint file is in the right hand tape drive.
   b. Enter: Waypoint file name
   c. Press: CONT

4. When "WAYPOINT?" appears on the display line:

   a. Enter: Waypoint of interest
   b. Press: CONT

5. When "Dx/Dy 1 OR RANGE/BEARING 2, 1 OR 2" appears on the display line:

   a. If the offset is to be applied in x and y components:
      (1) Enter: 1
      (2) Press: CONT
      (3) Go to Step 6
   b. If the offset is to be applied as a range and bearing from the waypoint:
      (1) Enter: 2
      (2) Press: CONT
      (3) Go to Step 8

6. When "INPUT DX (KM)" appears on the display line:

   a. Enter: x direction component of the offset in KM
   b. Press: CONT

7. When "INPUT DY (KM)" appears on the display line:

   a. Enter: y direction component of the offset in KM
   b. Press: CONT
   c. Go to Step 9

8. When "RANGE (KM) AND BEARING (DEG)" appears on the display line:

   a. Enter: The range in KM and the bearing in degrees separated by a comma
   b. Press: CONT

9. The effect of offsetting the waypoint on the TDs corresponding to the chain configuration will be printed on the hard copy printer.

10. When "APPLY CORRECTION TO WP? Y OR N" appears on the display line:

   a. If you want to change the waypoint parameters to the offset position:
      (1) Enter: Y
      (2) Press: CONT. (Note: This step does not change the waypoint data stored on magnetic tape. See K23.)
      (3) "WAYPOINT TD AND POSITION CORRECTED" is printed on the hard copy printer.
   b. If not:
      (1) Enter: N
      (2) Press: CONT

11. The program function is complete. The CRT is cleared and the program menu reprinted.

Function: Link Data Files

Special Function Key: K25

Subprograms:

Link (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
Store (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)

This program function enables multiple data files to be loaded into memory. Range and TD data can be corrected for each file entered. The total number of samples must be equal to or less than 400. The function will automatically ignore any samples which would cause this limit to be exceeded.

Function Variables:

W(*), X(*), Y(*), Z(*) - TD data arrays
R1(*), R2(*) - Miniranger data arrays
U$(*) - Time (Julian day:hours:min:sec) data array
N - Total number of samples
F$ - Final file name

User Instructions:

Prerequisite functions: none

1. Press K25. The program menu is cleared and "LINK DATA FILES" is printed on the CRT.

2. When "DATA FILE TYPE? TD & RANGE1 OR TD ONLY2" appears on the display line:

a. If the data file contains both TD and range data arrays (e.g. post Delaware data or reformatted Delaware data):
(1) Enter: 1
(2) Press: CONT
b. If the data file contains only TD arrays (e.g. Delaware and pre-Delaware data):
(1) Enter: 2
(2) Press: CONT

3. When "FIRST FILE NAME?" appears on the display line:

a. Enter: File name
b. Press: CONT

4. When "CORRECT TD DATA?" appears on the display line:

a. If TD data is to be corrected:
(1) Enter: Y
(2) Press: CONT
b. If not

(1) Enter: N
(2) Press: CONT
(3) Go to Step 11

5. The start and stop time for the data collected on the current data file are printed on the hard copy printer. When "CORRECTION TO TDW(MICROSEC)?" appears on the display line:

a. Enter: Correction to TDW in microsec.
b. Press: CONT

6. When "CORRECTION TO TDX(MICROSEC)?" appears on the display line:

a. Enter: Correction to TDX in microsec.
b. Press: CONT

7. When "CORRECTION TO TDY(MICROSEC)?" appears on the display line:

a. Enter: Correction to TDY in microsec.
b. Press: CONT

8. When "CORRECTION TO TDZ(MICROSEC)?" appears on the display line:

a. Enter: Correction to TDZ in microsec.
b. Press: CONT

The file name is annotated with a "t" and the corrections entered are printed on the hard copy printer.

9. A list of the files linked and total number of samples is printed on the CRT. When "ANOTHER FILE?" appears on the display line:

a. If another data file is to be entered:
   (1) Enter: Y
   (2) Press: CONT
b. If not
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 13

10. When "FILE NAME?" appears on the display line:

a. Enter: file name
b. Press: CONT
c. Go to Step 3

11. A final list of the files linked and total number of samples is printed on the hard copy printer. When "STORE DATA SET?" appears on the display line:

a. If you want to store the data on a new file
   (1) Enter: Y
   (2) Press: CONT

b. If not
(1) Enter: N
(2) Press: CONT
(3) Go to Step 15

12. When "FILE NAME?" appears on the display line

a. Insert a tape cartridge in the left hand tape drive.
b. Enter: file name
c. Press: CONT

13. If the data is not stored on tape, the file name will be the list of files linked. If the data was stored, the file name returned is the name used to store the data.

14. The program function is complete. The CRT is cleared and the program menu reprinted.

Function: Measured-Projected TDs

Special Function Key: K27

Subprograms:

Cart coord(Xmit(*), Wpt(25,7) Wpt(25,8) Zxmit(*))
Project (W, Wpt(*), W(*), X(*), Y(*), Z(*), Zx(*), Zy(*), Conf, Zxmit(*), N, V)
Rotate (Glat, Glon, R(*))
Cart (Plat, Plon, X, Y)
FNGlat (Lat, F)
Pseudo (Glat, Glon, R(*), Plat, Plon)
Rb (x1, y1, x2, y2, B, R)

This program function calculates the statistics of the difference between the TDs measured and TDs projected from the waypoint based on the xy position calculated from the measured TDs. The result is a measure of the three TD fix triangle. The function can be used to estimate the third TD of a waypoint if two are known.

Function variables:

C1 - status variable which indicates if chain data is in memory
F1- status variable which indicates if the waypoint table is in memory
Chain$ - chain file name
Ch$ - first four characters if Chain$
Conf - chain configuration, fifth character in Chain$
Xmit(*), Power(*), Emis(*) - transmitting station parameters: geodetic position, power level and emission delay
Wpt(*) - waypoint table
File$ - waypoint file name
Zxmit(*) - transmitter xy coordinates referenced to local origin
W - waypoint of interest
W(*), X(*), Y(*), Z(*) - TD data arrays
Pos_x(*), Pos_y *) - measurement position coordinates calculated from Miniranger data
N - Number of samples
V - Velocity of propagation
F$ - data file name

<u>User Instructions:</u>

Prerequisite functions:

K0 - Read Data
K16 - Edit Data (Optional)
K17 - Calculate XY Position from TD Data

1. Press K27. The CRT is cleared and "MEASURED-PROJECTED TDs" is printed on the CRT.

2. If "CHAIN FILE?" appears on the display line:

(1) Enter: Chain file name
(2) Press: CONT

3. If "WAYPOINT FILE NAME?" appears on the display line:

    a. Enter: waypoint file name
    b. Press: CONT

4. When "WAYPOINT FROM WHICH TDs ARE TO BE PROJECTED?" appears on the display line:

    a. Enter: waypoint number
    b. Press: CONT

5. The data file name is annotated with a " ". "RESULTS OF CALCULATING MEASURED TDS - PROJECTED TDS FROM WAYPOINT(W); FILE = (F$)" is printed on the hard copy printer followed by the statistics table generated by function K2.

6. The program function is complete. The CRT is cleared and the program menu is reprinted on the CRT.

Note: The data contained in the TD arrays (W(*), X(*), Y(*), Z(*)) has been changed.

Function: Simulate Waypoint Survey

Special Function Key: K28

Subprograms:

Bear(Wpt(WP-1,7), Wpt(Wp-1,8) Wpt(Wp,7), Wpt(Wp,8), Ca)
Data(W(*), X(*), Y(*), Z(*), N, Ca, Ds, De, Sigma, Xmit(*), Zp(*),
  V, Conf)
Stat_mat(W(*), X(*), Y(*), Z(*), Stat(*), Cov(*), O_set(*), N)
Reg(Stat(*), Cov(*), S1(*), Rone(*), O_set(*))
Print(Stat(*), O_set(*), Cov(*), Rone(*), N, 1)
Way(S1(*), S2(*), Rone(*), Rtwo(*), N, N, Wp(*))
Wprint(Wp(*))
Range (P1, P2, P3, P4, R)

This program simulates the survey of a waypoint using the intersection of visual ranges technique. Two TD data sets are generated along the tracklines defining a waypoint. Input variables are the waypoint of interest, the starting and stopping points of the survey lines, and the expected standard deviation of the TDs. The program function processes the data as if it were field data. Statistics and regression parameter tables are printed on the hard copy printer for each data set along with the waypoint table. The "true" TD values for the waypoint are zero. The tabulated waypoint value is the error in estimating the waypoint.

Function variables:

C1 - Indicates if Loran-C chain data has been read in memory
Chain$ - Loran-C chain file. First four characters are the abbreviation for the chain (e.g. NEUS, SEUS, GTLK, etc.) The fifth character is a number from 1 to 4 which designates the chain configuration: 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ
Ch$ - Chain name. First four characters of Chain$
Conf - Chain configuration
Xmit(*), Power(*), Emis(*) - Loran-C chain parameters, i.e. geodetic positions, power level, emission delays
F1 - Indicates if the waypoint file has been read into memory
Wpt(*) - Waypoint table
Wp - Waypoint of interest
Ca, Cb - Bearing from adjoining waypoints to waypoint of interest
Ds, De - Start and stop distances from waypoint
Sigma - Noise, i.e. expected standard deviation of time differences
N - Number of samples
Zp(*) - Waypoint position (latitude, longitude)
W(*), X(*), Y(*), Z(*) - TD data arrays
Stat(*), Cov(*), O_set(*) - Statistics arrays
S1(*), S2(*), Rone(*), Rtwo(*) - Statistics and regression summary arrays
V - Velocity of propagation
Wp(*) - Calculated waypoint table

User Instructions:

Prerequisite functions: A preliminary waypoint file containing estimated latitudes and longitudes for each waypoint is required. If a Loran-C chain other than NEUS1 is being used, a Loran-C chain file is necessary.

1. Press K28. The program menu is cleared and "SIMULATE WAYPOINT SURVEY" is printed on the CRT.

2. If "CHAIN FILE?" appears on the display line:

   (1) Enter: chain file name
   (2) Press: CONT

3. If "WAYPOINT FILE NAME?" appears on the display line:

   a. Enter: waypoint file name
   b. Press: CONT

4. When "WAYPOINT OF INTEREST?" appears on the display line:

   a. Enter: waypoint number
   b. Press: CONT

The bearings from the adjoining waypoints to the waypoint of interest will be calculated and the results printed on the hard copy printer.

5. When "START POINT RELATIVE TO WAYPOINT (KM)?" appears on the display line:

   a. Enter: Start point in KM. Both survey tracklines start the same distance from the waypoint. A positive start point indicates a position between the waypoints on the centerline. A negative start point indicates a position on the trackline beyond the waypoint of interest.
   b. Press: CONT

6. When "STOP POINT RELATIVE TO WAYPOINT (KM)" appears on the display line:

   a. Enter: Stop point in KM. Both survey tracklines end the same distance from the waypoint. A positive stop point indicates a position between the waypoints on the centerline. A negative stop point indicates a position on the centerline beyond the waypoint of interest.

   b. Press: CONT

Start and Stop points are printed on the hard copy printer.

7. When "EXPECTED STANDARD DEVIATION OF TD (MICROSEC)" appears on the display line:

   a. Enter: expected standard deviation of TDs in microseconds. All TDs are assigned the same standard deviation.

   b. Press: CONT

8. When "NUMBER OF SAMPLES PER TRACKLINE" appears on the display line:

a. Enter: number of samples. The number is the same for both survey tracklines.

b. Press: CONT

The expected standard deviation and number of samples is printed on the hard copy printer.

9. A data set is calculated for each trackline. The current sample being calculated and total number of samples appears on the display line. After a data set is complete, statistics and regression parameters are calculated and a summary table is printed on the hard copy printer. The sequence is repeated for the second trackline. The waypoint estimates, estimated rms error, and crossing angles are calculated and the results are printed on the hard copy printer. The "true" waypoint TDs are each zero. The values printed in the waypoint row are the estimation errors for the simulated survey.

10. The program function is complete.

TLS 1 SUBROUTINES

SUBROUTINE	CALLED FROM
Read	K0,21
Bear	K20,28
Read_alt	K0,21
Ct_at	K17
Way	K21,28
Dcom_isr	K29
Td	K20
G_mat	K17,20,22
Hi_lo	K2,5,18,19
Track	K17
Stat_mat	K2,17,21,28
Range	K20,28
Reg	K2,17,21,28
Patct	K19
Print	K2,21,28
Xplot	K18
Separate	K3
Wpf2	K17,22
Store	K3,16,25
Fehgt	K17,22
Plot	K4
Td2	K17
Rplot	K5
Td3	K17
Delete	K16
Fehg	K17,22
Delete_block	K16
Delete_td	K16
Wp2	K17,22
Cart_coord	K17,22,24,27
Wp3	K17,22
Rb	K17,22,24,27
Wprint	K21,28
Wpfile	K23
Link	K25
Project	K27
Data	K28
Rotate	K17,22,24,27
Pseudo	K17,22,24,27
Cart	K17,22,24,27

A-43 - A-44
PREVIOUS PAGES ARE BLANK

```
10   OPTION BASE 1
20    DEG
30   DIM W(400),X(400),Y(400),Z(400),R1(400),R2(400),U$(400)[14]
40   DIM P(4),O_set(4),Stat(4,4),Cov(3,6),S(4,6),R(3,6),O(4)
50    DIM S1(4,6),S2(4,6),Wp(4,6),Wpt(25,8),T(3),Zd(2),F$[80]
60    DIM Xmit(4,2),Emis(3),Bear(4),Range(4),Td(3),Power(4),G123(2,3),G12(2,2)
70    DIM G13(2,2),G23(2,2),Zxmit(4,2),Zp(2),Aa(3,2),Pos_x(400),Pos_y(400),Tp(3,
1),Tq(3,1)
80    DIM Zq(2,1),Zpp(2,1),At(400),Ct(400),Tpp(2,1),Tqq(2,1),Zqq(2,1)
90    DIM Zb(2),Rotate(3,3),Gx(4,2),Px(4,2),Axy(2,2)
100   DIM Tw(2),Zm(2),Zt(2)
110   DIM Zxx(2,1),Zxm(3,2),G(2,2),Rone(3,6),Rtwo(3,6)
120   COM A(400),B(400),C(400),D(400),R1t(400),R2t(400),Ut$(400)[14],T$[80]
150   V=.299792458/1.000338
160   F=.00335278
280 Keys:  !
290   ON KEY #0 GOTO K0
300   ON KEY #2 GOTO K2
310   ON KEY #3 GOTO K3
320   ON KEY #4 GOTO K4
330   ON KEY #5 GOTO K5
340   ON KEY #16 GOTO K16
350   ON KEY #17 GOTO K17
360   ON KEY #18 GOTO K18
370   ON KEY #19 GOTO K19
380   ON KEY #20 GOTO K20
390   ON KEY #21 GOTO K21
400   ON KEY #22 GOTO K22
410   ON KEY #23 GOTO K23
420   ON KEY #24 GOTO K24
430   ON KEY #25 GOTO K25
440   ON KEY #26 GOTO K26
450   ON KEY #27 GOTO K27
460   ON KEY #28 GOTO K28
480 Menu:  !
490   MASS STORAGE IS ":T15"
500   PRINTER IS 16
510   PRINT PAGE
520   PRINT "      VISUAL SURVEY: ANALYSIS 1    "
530   PRINT "K0:READ TRACKLINE FILE DATA"
540   PRINT "K2:STATS AND REGRESSION O  TD DATA"
550   PRINT "K3:SEPARATE DATA INTO SUBFILES"
560   PRINT "K4:PLOT TD DATA WITH REGRESSION LINE"
570   PRINT "K5:PLOT RESIDUALS OF TD DATA"
580   PRINT "K16:EDIT DATA"
590   PRINT "K17:CONVERT TDs TO XY AND AT/CT"
600   PRINT "K18:PLOT XY DATA"
610   PRINT "K19:PLOT/LIST AT/CT DATA"
620   PRINT "K20:PREDICT TD"
630   PRINT "K21:CALCULATE WAYPOINT"
640   PRINT "K22:DAISY CHAIN WAYPOINTS"
650   PRINT "K23:FILE OR READ WP DATA"
660   PRINT "K24:TD MOVE"
670   PRINT "K25:LINK DATA FILES"
```

```
680   PRINT "K26:STORE CHAIN DATA"
690   PRINT "K27:MEASURED-PROJECTED TDs"
700   PRINT "K28: SIMULATE WAYPOINT SURVEY"
720 Loop:  !
730   GOTO Loop
```

```
740 K0:! ^   READ DATA FILE
750     PRINT PAGE
760     PRINT "    READ DATA FILE    "
770     MASS STORAGE IS ":T14"
780     PRINTER IS 0
790     FIXED 2
840     CALL Read(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
870     PRINT "FILE=";F$;TAB(30);"SAMPLES=";N
880 Td_cor:   !
890     Td_cor$="N"
900     INPUT "CORRECT TDs? Y or N",Td_cor$
910     IF Td_cor$<>"Y" THEN Menu
920     PRINT "START TIME=";U$(1);TAB(30);"STOP TIME=";U$(N)
930     INPUT "CORRECTION TO TDW(MICROSEC)?",Wcor
940     INPUT "CORRECTION TO TDX(MICROSEC)?",Xcor
950     INPUT "CORRECTION TO TDY(MICROSEC)?",Ycor
960     INPUT "CORRECTION TO TDZ(MICROSEC)?",Zcor
970     FOR I=1 TO N
980     W(I)=W(I)+Wcor
990     X(I)=X(I)+Xcor
1000    Y(I)=Y(I)+Ycor
1010    Z(I)=Z(I)+Zcor
1020    NEXT I
1030    PRINT "TD CORRECTIONS:";TAB(20);"Wcor=";Wcor;TAB(35);"Xcor=";Xcor;TAB(50)
;"Ycor=";Ycor;TAB(65);"Zcor=";Zcor
1040    F$=F$&"t"
1050    GOTO Menu
1060    !
1070    !
```

```
1950 K2:! ^   STATS AND REGRESSION
1960   MASS STORAGE IS ":T14"
1970   PRINT PAGE
1980   PRINT "       STATISTICS AND REGRESSION OF TD DATA      "
1990   PRINTER IS 0
2000   CALL Hi_lo(W(*),N,Bw,Lw)
2010   CALL Hi_lo(X(*),N,Bx,Lx)
2020   CALL Hi_lo(Y(*),N,By,Ly)
2030   CALL Hi_lo(Z(*),N,Bz,Lz)
2040   CALL Stat_mat(W(*),X(*),Y(*),Z(*),Stat(*),Cov(*),O_set(*),N)
2050   CALL Reg(Stat(*),Cov(*),S(*),R(*),O_set(*))
2060   PRINT "TRACKLINE=";F$;TAB(20);" START TIME=";U$(1);TAB(50);" STOP TIME="
U$(N)
2070   CALL Print(Stat(*),O_set(*),Cov(*),R(*),N,1)
2080   GOTO Menu
2090  !
```

```
2100 K3: ! SEPARATE DATA INTO SUBFILES
2110  PRINTER IS 16
2120  PRINT PAGE
2130  PRINT "      SEPARATE DATA INTO SUBFILES     "
2140  MASS STORAGE IS ":T14"
2150  CALL Separate(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),A(*),B(*),C(*),D(*),R
1t(*),R2t(*),Ut$(*),Nt)
2160  CALL Store(A(*),B(*),C(*),D(*),R1t(*),R2t(*),Ut$(*),Nt,F$)
2161  INPUT "ANOTHER SEPARATION? Y or N",S$
2162  IF UPC$(S$[1,1])="Y" THEN K3
2170  GOTO Menu
2180  !
```

```
2200 K4:!  ^   PLOT TD DATA WITH REGRESSION
2210     PRINT PAGE
2220     PRINT "      PLOT TD DATA WITH REGRESSION LINES      "
2230     INPUT "PLOT? WX1,WY2,WZ3,XY4,XZ5,YZ6",Plot
2240     ON Plot GOTO Wx,Wy,Wz,Xy,Xz,Yz
2250 Wx:   PRINT LIN(3)
2260     CALL Plot(Lw,Bw,Lx,Bx,S(1,1),S(2,1),W(*),X(*),R(1,1),N,"TDX","TDW",F$)
2270     PRINT LIN(3)
2280     GOTO Menu
2290 Wy:    PRINT LIN(3)
2300     CALL Plot(Lw,Bw,Ly,By,S(1,2),S(2,2),W(*),Y(*),R(1,2),N,"TDY","TDW",F$)
2310     PRINT LIN(3)
2320     GOTO Menu
2330 Wz:    PRINT LIN(3)
2340     CALL Plot(Lw,Bw,Lz,Bz,S(1,3),S(2,3),W(*),Z(*),R(1,3),N,"TDZ","TDW",F$)
2350     PRINT LIN(3)
2360     GOTO Menu
2370 Xy:   PRINT LIN(3)
2380     CALL Plot(Lx,Bx,Ly,By,S(1,4),S(2,4),X(*),Y(*),R(1,4),N,"TDY","TDX",F$)
2390     PRINT LIN(3)
2400     GOTO Menu
2410 Xz:   PRINT LIN(3)
2420     CALL Plot(Lx,Bx,Lz,Bz,S(1,5),S(2,5),X(*),Z(*),R(1,5),N,"TDZ","TDX",F$)
2430     PRINT LIN(3)
2440     GOTO Menu
2450 Yz:   PRINT LIN(3)
2460     CALL Plot(Ly,By,Lz,Bz,S(1,6),S(2,6),Y(*),Z(*),R(1,6),N,"TDZ","TDY",F$)
2470     GOTO Menu
2500  !
```

```
2510 KS:! ^   PLOT RESIDUALS
2520    PRINT PAGE
2530    Hc=0
2540    PRINT "     PLOT RESIDUALS     "
2550    INPUT "PLOT? WX1,WY2,WZ3,XY4,XZ5,YZ6",Plot
2560    ON Plot GOTO Rwx,Rwy,Rwz,Rxy,Rxz,Ryz
2570 Rwx:  IF R(3,1)=2 THEN Skip1
2580    CALL Rplot(W(*),X(*),S(*),R(*),1,N,"TDW","TDX",F$)
2590    PRINT LIN(3)
2600    GOTO Menu
2610 Skip1:CALL Rplot(X(*),W(*),S(*),R(*),1,N,"TDW","TDX",F$)
2620    PRINT LIN(3)
2630    GOTO Menu
2640 Rwy:  IF R(3,2)=2 THEN Skip2
2650    CALL Rplot(W(*),Y(*),S(*),R(*),2,N,"TDW","TDY",F$)
2660    PRINT LIN(3)
2670    GOTO Menu
2680 Skip2:CALL Rplot(Y(*),W(*),S(*),R(*),2,N,"TDW","TDY",F$)
2690    PRINT LIN(3)
2700    GOTO Menu
2710 Rwz:  IF R(3,3)=2 THEN Skip3
2720    CALL Rplot(W(*),Z(*),S(*),R(*),3,N,"TDW","TDZ",F$)
2730    PRINT LIN(3)
2740    GOTO Menu
2750 Skip3:CALL Rplot(Z(*),W(*),S(*),R(*),3,N,"TDW","TDZ",F$)
2760    PRINT LIN(3)
2770    GOTO Menu
2780 Rxy:  IF R(3,4)=2 THEN Skip4
2790    CALL Rplot(X(*),Y(*),S(*),R(*),4,N,"TDX","TDY",F$)
2800    PRINT LIN(3)
2810    GOTO Menu
2820 Skip4:CALL Rplot(Y(*),X(*),S(*),R(*),4,N,"TDX","TDY",F$)
2830    PRINT LIN(3)
2840    GOTO Menu
2850 Rxz:  IF R(3,5)=2 THEN Skip5
2860    CALL Rplot(X(*),Z(*),S(*),R(*),5,N,"TDX","TDZ",F$)
2870    PRINT LIN(3)
2880    GOTO Menu
2890 Skip5:CALL Rplot(Z(*),X(*),S(*),R(*),5,N,"TDX","TDZ",F$)
2900    PRINT LIN(3)
2910    GOTO Menu
2920 Ryz:  IF R(3,6)=2 THEN Skip6
2930    CALL Rplot(Y(*),Z(*),S(*),R(*),6,N,"TDY","TDZ",F$)
2940    PRINT LIN(3)
2950    GOTO Menu
2960 Skip6:CALL Rplot(Z(*),Y(*),S(*),R(*),6,N,"TDY","TDZ",F$)
2970    PRINT LIN(3)
2980    GOTO Menu
3000    !
```

```
3020 K16:!  ^    EDIT TD DATA
3030      MASS STORAGE IS ":T14"
3040     PRINT PAGE
3050     PRINT "        EDIT TD DATA      "
3051     Edit=4
3060     INPUT "SINGLE LINE1,BLOCK2,OR TD-CLIP3?",Edit
3070     ON Edit GOTO Single,Blok,Clip,More
3080 Single:CALL Delete(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
3090     GOTO More
3100 Blok:CALL Delete_blok(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
3110     GOTO More
3120 Clip:CALL Delete_td(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
3121 More: !
3122  INPUT "EDIT MORE DATA? (Y or N)",S$
3123  IF UPC$(S$[1,1])="Y" THEN K16
3124  INPUT "LIST DATA?",S$
3125  IF UPC$(S$[1,1])<>"Y" THEN Stor
3126  FIXED 2
3127  FOR I=1 TO N
3128  PRINT I;W(I);X(I);Y(I);Z(I);R1(I);R2(I);U$(I)
3129  NEXT I
3140 Stor: !
3150     INPUT "  STORE EDITED DATA? Y OR N  ",S$
3160     IF UPC$(S$[1,1])<>"Y" THEN Menu
3170     CALL Store(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
3180     GOTO Menu
3210     !
```

```
3220 K17:! ^   CONVERT TD TO XY
3230      PRINT PAGE
3240      PRINT "     CONVERT TD TO XY     "
3250      MASS STORAGE IS ":T15"
3260      MAT Pos_x=(0)
3270      MAT Pos_y=(0)
3280      MAT Ct=(0)
3290      MAT At=(0)
3300 Chain:     !
3310      IF C1=1 THEN Wp_file
3330      INPUT "CHAIN FILE?",Chain$
3340      C1=1
3360      ASSIGN #1 TO Chain$
3370      READ #1;Xmit(*),Power(*),Emis(*)
3380 Wp_file:     !
3390      IF F1=1 THEN Xmit_coord
3400      INPUT "WAYPOINT FILE NAME?",File$
3410      F1=1
3420      ASSIGN #1 TO File$
3430      READ #1;Wpt(*)
3440 Xmit_coord:     !
3450      CALL Cart_coord(Xmit(*),Wpt(25,7),Wpt(25,8),Zxmit(*))
3460 Waypoint:     !
3470      INPUT "WAYPOINT TO",Wt
3480      INPUT "WAYPOINT FROM",Wf
3490 Range_bearing:     !
3500      FOR I=1 TO 4
3510      CALL Rb(Wpt(Wt,5),Wpt(Wt,6),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I))
3520      NEXT I
3530 G_mat:     !
3540      Zp(1)=Wpt(Wt,5)
3550      Zp(2)=Wpt(Wt,6)
3560      CALL G_mat(Power(*),V,Range(*),Bear(*),Zp(*),Aa(*),G123(*),G12(*),G23(*
),G13(*))
3570 Configuration:     !
3580      PRINTER IS 0
3600      Conf=VAL(Chain$[5,5])
3610      Ch$=Chain$[1,4]
3630      INPUT "TWO OR THREE TD SOLUTION?",So
3640      IF So=2 THEN So2
3650      IF So=3 THEN So3
3660 So3:     ! THREE TD SOLUTION
3670      CALL Wp3(Wpt(*),Conf,Wt,Tp(*),Zpp(*),L$)!WAYPOINT TD
3680      PRINT "THREE TD SOLUTION, CHAIN=";Ch$;"   LOPs=";L$
3690      FOR I=1 TO N
3700      CALL Td3(W(*),X(*),Y(*),Z(*),I,Conf,Tq(*))
3710      CALL Fehg(Zxmit(*),Zpp(*),Tp(*),Tq(*),G123(*),Zq(*),V,3)
3720      Pos_x(I)=Zq(1,1)
3730      Pos_y(I)=Zq(2,1)
3740      DISP I
3750      NEXT I
3760      ! CALCULATE POSITION OF FROM WAYPOINT (Wf)
3770      CALL Wp3(Wpt(*),Conf,Wf,Tq(*),Zxx(*),L$)!WAYPOINT FROM TD
3780      CALL Fehg(Zxmit(*),Zpp(*),Tp(*),Tq(*),G123(*),Zq(*),V,3)
```

```
5790     GOTO Ct_at
5800 So2:     ! TWO TD SOLUTION
5810     CALL Wp2(Wpt(*),Conf,Wt,Tpp(*),Pair,Zpp(*),G12(*),G23(*),G13(*),G(*),L$
,Zxmit(*),Zxm(*))
5820     PRINT "TWO TD SOLUTION, CHAIN=";Ch$;"   LOPs=";L$
5830     FOR I=1 TO N
5840     CALL Td2(W(*),X(*),Y(*),Z(*),I,Conf,Pair,Tqq(*))
5850     CALL Fehg(Zxm(*),Zpp(*),Tpp(*),Tqq(*),G(*),Zq(*),V,2)
5860     Pos_x(I)=Zq(1,1)
5870     Pos_y(I)=Zq(2,1)
5880     DISP I
5890     NEXT I
5900     ! CALCULATE POSITION OF FROM WAYPOINT(Wf)
5910     CALL Wpf2(Wpt(*),Conf,Pair,Wf,Tqq(*))
5920     CALL Fehg(Zxm(*),Zpp(*),Tpp(*),Tqq(*),G(*),Zq(*),V,2)
5930     ! CALCULATE CROSSTRACK AND ALONGTRACK POSITIONS
5940 Ct_at:  CALL Ct_at(Pos_x(*),Pos_y(*),Ct(*),At(*),Wpt(Wt,5),Wpt(Wt,6),Zq(1,1
,Zq(2,1),N,R,Angle)
5950     DISP R,Angle
5960     CALL Stat_mat(Pos_x(*),Pos_y(*),Ct(*),At(*),Stat(*),Cov(*),O_set(*),N)
5970     CALL Reg(Stat(*),Cov(*),S(*),R(*),O_set(*))
5980     PRINTER IS 0
5990  CALL Track(Stat(*),O_set(*),Cov(*),R(*),Sample,F$,Wt,Wf,Angle,N)
6000     GOTO Menu
6020     !
```

```
4030 K18:!  ^   PLOT XY DATA
4040      PRINT PAGE
4041      Plot=1
4050      PRINT "      PLOT XY DATA      "
4051      INPUT "STANDARD XY PLOT1 OR PLOT TO CHART SCALE2? 1 OR 2",Plot
4052      IF Plot=2 THEN Chart
4060      CALL Hi_lo(Pos_x(*),N,Maxx,Minx)
4070      CALL Hi_lo(Pos_y(*),N,Maxy,Miny)
4080      CALL Xplot(1,Pos_x(*),Pos_y(*),Minx,Maxx,Miny,Maxy,N,1,F$,Ch$,L$,2)
4090      GOTO Menu
4091 Chart:     !
4092      INPUT "WAYPOINT FOR AXES?",Axes
4093      Xaxes=Wpt(Axes,5)
4094      Yaxes=Wpt(Axes,6)
4095      INPUT "WAYPOINT OFFSET FROM LOWER LEFT CORNER OF PLOT? X,Y(KM)",Xoffset
,Yoffset
4096      CALL Xplot(1,Pos_x(*),Pos_y(*),Xaxes,Xoffset,Yaxes,Yoffset,N,1,F$,Ch$,L
$,5)
4097      GOTO Menu
4100  !
```

```
4110 KEY 19 ! ^  PLOT CT/AT DATA
4120     PRINT PAGE
4130     PRINT "     PLOT/LIST ALONG TRACK CROSS TRACK DATA    "
4131 Opt: INPUT "PLOT 1; LIST 2; QUIT 3",Opt
4132     IF (Opt<1) OR (Opt>3) THEN Opt
4133     ON Opt GOTO Plot,List,End
4140 Plot: CALL Hi_lo(Ct(*),N,Maxc,Minc)
4150     CALL Hi_lo(At(*),N,Maxa,Mina)
4160     CALL Patct(At(*),Ct(*),N,Mina,Maxa,Minc,Maxc,Wt,Wf,F$,R,1,Data)
4161     GOTO Opt
4163 List:    PRINTER IS 0
4164     PRINT "FILE:  ";F$
4165     PRINT LIN(1);"SAMPLE";TAB(15);"ALONG TRACK (KM)";TAB(34);"CROSS TRACK (
M)";TAB(55);"TIME (D:H:M:S)"
4166     PRINT LIN(1)
4167     FOR I=1 TO N
4168     PRINT I;TAB(20);At(I);TAB(37);Ct(I)*1000;TAB(55);U$(I)
4169     NEXT I
4170     GOTO Opt
4171 End:    GOTO Menu
4180     !
```

```
4200 K20:!  ^  PREDICT TD
4210      PRINT PAGE
4220      PRINT "       TD PREDICTION     "
4230      MASS STORAGE IS ":T15"
4240      PRINTER IS 0
4250      FIXED 3
4260      IF C1=1 THEN Wpfile
4280      INPUT "CHAIN FILE?",Chain$
4290      C1=1
4310      ASSIGN #1 TO Chain$
4320      READ #1;Xmit(*),Power(*),Emis(*)
4330 Wpfile: INPUT "IS POINT OF INTEREST ON WAYPOINT FILE? Y OR N",P$
4340      IF P$="N" THEN Input
4350      IF F1=1 THEN Jump
4360      INPUT "WP FILE NAME?",File$
4370      F1=1
4380      ASSIGN #1 TO File$
4390      READ #1;Wpt(*)
4400 Jump:      INPUT "WAYPOINT NO.?",Wn
4410      Lat=Wpt(Wn,7)
4420      Lon=Wpt(Wn,8)
4430      GOTO Td
4440 Input:     INPUT " INPUT LAT IN DEGREES,MIN,SEC",D,M,S
4450      Lat=D+M/60+S/3600
4460      INPUT "INPUT LON IN DEGREES,MIN,SEC",D,M,S
4470      Lon=-(D+M/60+S/3600)
4480 Td:    FOR I=1 TO 4
4490      CALL Range(Lat,Lon,Xmit(I,1),Xmit(I,2),Range(I))
4500      CALL Bear(Lat,Lon,Xmit(I,1),Xmit(I,2),Bear(I))
4510      NEXT I
4520      CALL Td(Range(*),V,Emis(*),Td(*))
4530      CALL G_mat(Power(*),V,Range(*),Bear(*),Zp(*),Aa(*),G123(*),G12(*),G23(*
),G13(*))
4540 Gdop:    !
4550      Gdop12=(G12(1,1)^2+G12(1,2)^2+G12(2,1)^2+G12(2,2)^2)^.5
4560      Gdop13=(G13(1,1)^2+G13(1,2)^2+G13(2,1)^2+G13(2,2)^2)^.5
4570      Gdop23=(G23(1,1)^2+G23(1,2)^2+G23(2,1)^2+G23(2,2)^2)^.5
4580      Gdop123=(G123(1,1)^2+G123(1,2)^2+G123(1,3)^2+G123(2,1)^2+G123(2,2)^2+G1
23(2,3)^2)^.5
4590      FIXED 4
4610      Ch$=Chain$[1,4]
4620      PRINT "CHAIN=";Ch$
4630      PRINT "LAT=";Lat;TAB(15);"LON=";Lon
4640      FIXED 3
4660      Conf=VAL(Chain$[5,5])
4670      ON Conf GOTO Txyz,Twxy,Twxz,Twyz
4680 Txyz:PRINT "PREDICTED: TDX=";Td(1);TAB(35);"TDY=";Td(2);TAB(55);"TDZ=";Td(3
)
4690      IF P$="N" THEN 4710
4700      PRINT "SURVEYED:  TDX=";Wpt(Wn,2);TAB(35);"TDY=";Wpt(Wn,3);TAB(55);"TDZ
=";Wpt(Wn,4)
4710      PRINT TAB(20);"RANGE(KM)";TAB(40);"BEARING(DEG)"
4720      PRINT "MASTER";TAB(20);Range(4);TAB(40);Bear(4)
4730      PRINT "X-RAY";TAB(20);Range(1);TAB(40);Bear(1)
```

```
4740     PRINT "YANKEE";TAB(20);Range(2);TAB(40);Bear(2)
4750     PRINT "ZULU";TAB(20);Range(3);TAB(40);Bear(3)
4760    PRINT TAB(22);"XYZ";TAB(37);"XY";TAB(52);"XZ";TAB(67);"YZ"
4770    PRINT "GDOP(M/NANOSEC)";TAB(20);Gdop123;TAB(35);Gdop12;TAB(50);Gdop13;TA
B(65);Gdop23
4780     GOTO Menu
4790 Twxy:PRINT "PREDICTED: TDW=";Td(1);TAB(35);"TDX=";Td(2);TAB(55);"TDY=";Td(3
)
4800     IF P$="N" THEN 4820
4810     PRINT "SURVEYED:  TDW=";Wpt(Wn,1);TAB(35);"TDX=";Wpt(Wn,2);TAB(55);"TDY
=";Wpt(Wn,3)
4820     PRINT TAB(20);"RANGE(KM)";TAB(40);"BEARING(DEG)"
4830     PRINT "MASTER";TAB(20);Range(4);TAB(40);Bear(4)
4840     PRINT "WHISKEY";TAB(20);Range(1);TAB(40);Bear(1)
4850     PRINT "XRAY";TAB(20);Range(2);TAB(40);Bear(2)
4860     PRINT "YANKEE";TAB(20);Range(3);TAB(40);Bear(3)
4870     PRINT TAB(22);"WXY";TAB(37);"WX";TAB(52);"WY";TAB(67);"XY"
4880     PRINT "GDOP(M/NANOSEC)";TAB(20);Gdop123;TAB(35);Gdop12;TAB(50);Gdop13;T
AB(65);Gdop23
4890     GOTO Menu
4900 Twxz:PRINT "PREDICTED: TDW=";Td(1);TAB(35);"TDX=";Td(2);TAB(55);"TDZ=";Td(3
)
4910     IF P$="N" THEN 4930
4920     PRINT "SURVEYED:  TDW=";Wpt(Wn,1);TAB(35);"TDX=";Wpt(Wn,2);TAB(55);"TDZ
=";Wpt(Wn,4)
4930     PRINT TAB(20);"RANGE(KM)";TAB(40);"BEARING(DEG)"
4940     PRINT "MASTER";TAB(20);Range(4);TAB(40);Bear(4)
4950     PRINT "WHISKEY";TAB(20);Range(1);TAB(40);Bear(1)
4960     PRINT "XRAY";TAB(20);Range(2);TAB(40);Bear(2)
4970     PRINT "ZULU";TAB(20);Range(3);TAB(40);Bear(3)
4980     PRINT TAB(22);"WXZ";TAB(37);"WX";TAB(52);"WZ";TAB(67);"XZ"
4990     PRINT "GDOP(M/NANOSEC)";TAB(20);Gdop123;TAB(35);Gdop12;TAB(50);Gdop13;T
AB(65);Gdop23
5000     GOTO Menu
5010 Twyz:PRINT "PREDICTED: TDW=";Td(1);TAB(35);"TDY=";Td(2);TAB(55);"TDZ=";Td(3
)
5020     IF P$="N" THEN 5040
5030     PRINT "SURVEYED:  TDW=";Wpt(Wn,1);TAB(35);"TDY=";Wpt(Wn,3);TAB(55);"TDZ
=";Wpt(Wn,4)
5040     PRINT TAB(20);"RANGE(KM)";TAB(40);"BEARING(DEG)"
5050     PRINT "MASTER";TAB(20);Range(4);TAB(40);Bear(4)
5060     PRINT "WHISKEY";TAB(20);Range(1);TAB(40);Bear(1)
5070     PRINT "YANKEE";TAB(20);Range(2);TAB(40);Bear(2)
5080     PRINT "ZULU";TAB(20);Range(3);TAB(40);Bear(3)
5090     PRINT TAB(22);"WYZ";TAB(37);"WY";TAB(52);"WZ";TAB(67);"YZ"
5100     PRINT "GDOP(M/NANOSEC)";TAB(20);Gdop123;TAB(35);Gdop12;TAB(50);Gdop13;T
AB(65);Gdop23
5110     GOTO Menu
5130     !
```

```
5140 K21:!  ^   CALCULATE WAYPOINT
5150     PRINTER IS 16
5160     PRINT PAGE
5170     PRINT "         CALCULATE WAYPOINT         "
5180     MASS STORAGE IS ":T14"
5190     PRINTER IS 0
5200     !           TYPE OF DATA FILE
5210     Read=1
5220     INPUT "DATA FILE TYPE? TD&RANGE1 OR TD ONLY2",Read
5230     IF (Read<>1) AND (Read<>2) THEN 5220
5240     !           READ FIRST DATA FILE AND CALCULATE STATS
5250     ON Read GOTO 5260,5280
5260     CALL Read(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N1,F$)
5270     GOTO 5290
5280     CALL Read_alt(W(*),X(*),Y(*),Z(*),U$(*),N1,F$)
5290     PRINT "TRACKLINE ONE=",F$
5300     CALL Stat_mat(W(*),X(*),Y(*),Z(*),Stat(*),Cov(*),O_set(*),N1)
5310     CALL Reg(Stat(*),Cov(*),S1(*),Rone(*),O_set(*))
5320     CALL Print(Stat(*),O_set(*),Cov(*),Rone(*),N1,1)
5330     !           READ SECOND DATA FILE AND CALCULATE STATS
5340     ON Read GOTO 5350,5370
5350     CALL Read(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N2,F$)
5360     GOTO 5380
5370     CALL Read_alt(W(*),X(*),Y(*),Z(*),U$(*),N2,F$)
5380     PRINT "TRACKLINE TWO=",F$
5390     CALL Stat_mat(W(*),X(*),Y(*),Z(*),Stat(*),Cov(*),O_set(*),N2)
5400     CALL Reg(Stat(*),Cov(*),S2(*),Rtwo(*),O_set(*))
5410     CALL Print(Stat(*),O_set(*),Cov(*),Rtwo(*),N2,1)
5420     !          CALCULATE INTERSECTION OF REGRESSION LINES
5430     CALL Way(S1(*),S2(*),Rone(*),Rtwo(*),N1,N2,Wp(*))
5440     CALL Wprint(Wp(*))
5450     GOTO Menu
5460      !
```

```
5470 K22:! ^   DAISY CHAIN
5480    PRINT PAGE
5490    PRINT "      DAISY CHAIN     "
5500    PRINTER IS 0
5510    MASS STORAGE IS ":T15"
5520    !       INPUT CHAIN DATA
5530    IF Ci=1 THEN 5610
5550    INPUT "CHAIN FILE?",Chain$
5560    Ci=1
5580    ASSIGN #1 TO Chain$
5590    READ #1;Xmit(*),Power(*),Emis(*)
5610    Conf=VAL(Chain$[5,5])
5620    Ch$=Chain$[1,4]
5630    !        INPUT WAYPOINT FILE DATA
5640    IF Fi=1 THEN 5690
5650    INPUT "WP FILE?",File$
5660    Fi=1
5670    ASSIGN #1 TO File$
5680    READ #1;Wpt(*)
5690    !        CALCULATE XY COORDINATES OF TRANSMITTERS
5700    CALL Cart_coord(Xmit(*),Wpt(25,7),Wpt(25,8),Zxmit(*))
5710    !        INPUT START AND STOP WAYPOINTS
5720    INPUT "START WAYPOINT NO.?",Rw
5730    INPUT "END WAYPOINT ?",Iw
5740    !        CALCULATE G-MATRICES FOR Rw
5750    FOR I=1 TO 4
5760    CALL Rb(Wpt(Rw,5),Wpt(Rw,6),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I))
5770    NEXT I
5780    Zp(1)=Wpt(Rw,5)
5790    Zp(2)=Wpt(Rw,6)
5800    CALL G_mat(Power(*),V,Range(*),Bear(*),Zp(*),Aa(*),G123(*),G12(*),G23(*)
,G13(*))
5810    !        INPUT TWO OR THREE TD SOLUTION
5820    INPUT "TWO OR THREE TD SOLUTION? 2 OR 3",So
5830    IF (So<2) OR (So>3) THEN 5820
5840    IF So=2 THEN Ctwo
5850    IF So=3 THEN Cthree
5860    !        CALCULATE POSITION OF Iw
5870 Ctwo:   !       TWO TD SOLUTION
5880     CALL Wp2(Wpt(*),Conf,Rw,Tpp(*),Pair,Zpp(*),G12(*),G23(*),G13(*),G(*),L$
,Zxmit(*),Zxm(*))
5890     PRINT "TWO TD SOLUTION, CHAIN=";Ch$;"  LOPs=";L$
5900     CALL Wpf2(Wpt(*),Conf,Pair,Iw,Tqq(*))
5910     CALL Fehg(Zxm(*),Zpp(*),Tpp(*),Tqq(*),G(*),Zq(*),V,2)
5920     GOTO Print
5930 Cthree: !       THREE TD SOLUTION
5940     CALL Wp3(Wpt(*),Conf,Rw,Tp(*),Zpp(*),L$)
5950     PRINT "THREE TD SOLUTION, CHAIN=";Ch$;"  LOPs=";L$
5960     CALL Wp3(Wpt(*),Conf,Iw,Tq(*),Zxx(*),L$)
5970     CALL Fehg(Zxmit(*),Zpp(*),Tp(*),Tq(*),G123(*),Zq(*),V,3)
5980 Print:!     PRINT RESULTS
5990     DEG
6000     Dx=Zq(1,1)-Wpt(Rw,5)
6010     Dy=Zq(2,1)-Wpt(Rw,6)
```

```
6020      Dlat=Dy/(1.852*60)
6030      Dlon=Dx/(1.852*60*COS(Wpt(Rw,7)))
6040      Lat=Wpt(Rw,7)+Dlat
6050      Lon=Wpt(Rw,8)+Dlon
6060      A=ATN(Dx/Dy)
6070      IF Dy<0 THEN A=A+180
6080      IF A<0 THEN A=A+360
6090      R=SQR(Dx^2+Dy^2)
6100      FIXED 0
6110      PRINT "LOCAL GRID CENTERED AT WP";25
6120      PRINT " WAYPOINT";Iw;"REFERENCED TO";Rw
6130      FIXED 3
6140      PRINT TAB(10);"Dx=";Dx,"Dy=";Dy;TAB(40);"D_lat=";Dlat,"D_lon=";Dlon
6150      PRINT TAB(10);"ANGLE=";A;"   RANGE=";R;"KM(";R/1.852;"NM,";R/.9144;"KYD
S)"
6160      PRINT TAB(10);"X=";Zq(1,1),"Y=";Zq(2,1);TAB(40);"LAT=";Lat,"LON=";Lon
6170      PRINT "CURRENT CO-ORDINATES:"
6180      PRINT TAB(10);"X=";Wpt(Iw,5),"Y=";Wpt(Iw,6);TAB(40);"LAT=";Wpt(Iw,7),"L
ON=";Wpt(Iw,8)
6190      INPUT "CHANGE POSITION AND LAT/LON OF WAYPOINT OF INTEREST? Y OR N",Ch$
6200      IF Ch$<>"Y" THEN 6260
6210      Wpt(Iw,5)=Zq(1,1)
6220      Wpt(Iw,6)=Zq(2,1)
6230      Wpt(Iw,7)=Lat
6240      Wpt(Iw,8)=Lon
6250      PRINT "WAYPOINT TABLE CHANGED"
6260      GOTO Menu
6270   !
```

```
6280 K23:! ^   WAYPOINT FILE
6290  PRINT PAGE
6300  PRINT "      FILE WAYPOINT DATA      "
6310  MASS STORAGE IS ":T15"
6320  CALL Wpfile(Wpt(*),Fi,File$)
6330  GOTO Menu
6350  !
```

```
6360 K24:! ^   TD MOVE

6370     PRINT PAGE
6380     PRINT "    MOVE: CALCULATES Tdq=Tdp+h(Zq)-h(Zp)    "
6390     PRINTER IS 0
6400     MASS STORAGE IS ":T15"
6410     FIXED 3
6420     !             INPUT CHAIN DATA
6430     IF C1=1 THEN 6510
6450     INPUT "CHAIN FILE?",Chain$
6460     C1=1
6480     ASSIGN #1 TO Chain$
6490     READ #1;Xmit(*),Power(*),Emis(*)
6510     Ch$=Chain$[1,4]
6520     Conf=VAL(Chain$[5,5])
6530     !             INPUT WAYPOINT FILE DATA
6540     IF F1=1 THEN 6590
6550     INPUT "WAYPOINT FILE?",File$
6560     F1=1
6570     ASSIGN #1 TO File$
6580     READ #1;Wpt(*)
6590     !            INPUT WAYPOINT NUMBER AND OFFSET
6600     INPUT "WAYPOINT?",Wp
6610     INPUT "Dx/Dy1 OR Range/Bearing2, 1 OR 2",Move
6620     IF (Move<>1) AND (Move<>2) THEN 6610
6630     ON Move GOTO X_y,Rb
6640 X_y:   INPUT "INPUT Dx(KM)",Dx
6650     INPUT "INPUT Dy(KM)",Dy
6660     GOTO 6700
6670 Rb:  INPUT "RANGE(KM) AND BEARING(DEG)",R,B
6680     Dx=R*SIN(B)
6690     Dy=R*COS(B)
6700     !             CALCULATE XY COORDINATES OF TRANSMITTERS
6710     CALL Cart_coord(Xmit(*),Wpt(25,7),Wpt(25,8),Zxmit(*))
6720     !             CALCULATE h(Zp)
6730     Zp(1)=Wpt(Wp,5)
6740     Zp(2)=Wpt(Wp,6)
6750     FOR I=1 TO 4
6760     CALL Rb(Zp(1),Zp(2),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I))
6770     NEXT I
6780     FOR I=1 TO 3
6790     T(I)=(Range(I)-Range(4))/V
6800     NEXT I
6810     !             CALCULATE Dtd=h(Zq)-h(Zp)
6820     Zp(1)=Zp(1)+Dx
6830     Zp(2)=Zp(2)+Dy
6840     FOR I=1 TO 4
6850     CALL Rb(Zp(1),Zp(2),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I))
6860     NEXT I
6870     FOR I=1 TO 3
6880     T(I)=(Range(I)-Range(4))/V-T(I)
6890     NEXT I
6900     !             PRINT RESULTS AND APPLY CORRECTION TO WP(OPTIONAL)
6910     FIXED 3
6920     PRINT "AT WAYPOINT:";Wp;", A MOVE OF Dx=";Dx;"KM AND Dy=";Dy;"KM"
```

```
6930     PRINT "RESULTS IN TD CHANGES(MICROSEC) OF:"
6940     ON Conf GOTO Dxyz,Dwxy,Dwxz,Dwyz
6950 Dxyz:PRINT TAB(5);"Dtdx=";T(1);TAB(20);"Dtdy=";T(2);TAB(40);"Dtdz=";T(3)
6960     INPUT "APPLY CORRECTION TO WP? Y OR N",Cor$
6970     IF Cor$<>"Y" THEN Menu
6980     Wpt(Wp,2)=Wpt(Wp,2)+T(1)
6990     Wpt(Wp,3)=Wpt(Wp,3)+T(2)
7000     Wpt(Wp,4)=Wpt(Wp,4)+T(3)
7010     Wpt(Wp,5)=Zp(1)
7020     Wpt(Wp,6)=Zp(2)
7030     PRINT "WAYPOINT TD AND POSITION CORRECTED"
7040     GOTO Menu
7050 Dwxy:PRINT TAB(5);"Dtdw=";T(1);TAB(20);"Dtdx=";T(2);TAB(40);"Dtdy=";T(3)
7060     INPUT "APPLY CORRECTION TO WP? Y OR N",Cor$
7070     IF Cor$<>"Y" THEN Menu
7080     Wpt(Wp,1)=Wpt(Wp,1)+T(1)
7090     Wpt(Wp,2)=Wpt(Wp,2)+T(2)
7100     Wpt(Wp,3)=Wpt(Wp,3)+T(3)
7110     Wpt(Wp,5)=Zp(1)
7120     Wpt(Wp,6)=Zp(2)
7130     PRINT "WAYPOINT TD AND POSITION CORRECTED"
7140     GOTO Menu
7150 Dwxz:PRINT TAB(5);"Dtdw=";T(1);TAB(20);"Dtdx=";T(2);TAB(40);"Dtdz=";T(3)
7160     INPUT "APPLY CORRECTION TO WP? Y OR N",Cor$
7170     IF Cor$<>"Y" THEN Menu
7180     Wpt(Wp,1)=Wpt(Wp,1)+T(1)
7190     Wpt(Wp,2)=Wpt(Wp,2)+T(2)
7200     Wpt(Wp,4)=Wpt(Wp,4)+T(3)
7210     Wpt(Wp,5)=Zp(1)
7220     Wpt(Wp,6)=Zp(2)
7230     PRINT "WAYPOINT TD AND POSITION CORRECTED"
7240     GOTO Menu
7250 Dwyz:PRINT TAB(5);"Dtdw=";T(1);TAB(20);"Dtdy=";T(2);TAB(40);"Dtdz=";T(3)
7260     INPUT "APPLY CORRECTION TO WP? Y OR N",Cor$
7270     IF Cor$<>"Y" THEN Menu
7280     Wpt(Wp,1)=Wpt(Wp,1)+T(1)
7290     Wpt(Wp,3)=Wpt(Wp,3)+T(2)
7300     Wpt(Wp,4)=Wpt(Wp,4)+T(3)
7310     Wpt(Wp,5)=Zp(1)
7320     Wpt(Wp,6)=Zp(2)
7330     PRINT "WAYPOINT TD AND POSITION CORRECTED"
7340     GOTO Menu
7360     !
```

```
7361 K25:! ^     LINK DATA FILES
7380    PRINT PAGE
7390    PRINT "        LINK TD DATA FILES        "
7400    MAT W=(0)
7410    MAT X=(0)
7420    MAT Y=(0)
7430    MAT Z=(0)
7440    FOR I=1 TO 400
7450    U$=""
7460    NEXT I
7470    CALL Link(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
7480    GOTO Menu
7481    !
```

7490 K26:! ^ STORE CHAIN DATA

```
7500   MASS STORAGE IS ":T15"
7510   PRINTER IS 16
7520   PRINT PAGE
7530   PRINT "   STORE CHAIN DATA   "
7540   PRINT LIN(10)
7550   PRINT "PLACE TAPE CARTRIDGE IN RIGHT HAND TAPE DRIVE"
7560   !        INPUT FILE NAME AND CHECK FOR CORRECT FORMAT
7570   INPUT "INPUT CHAIN FILE NAME",Chain$
7580   IF LEN(Chain$)<>5 THEN 7570
7590   IF (VAL(Chain$[5,5])>4) OR (VAL(Chain$[5,5])<1) THEN 7570
7600   Conf=VAL(Chain$[5,5])
7610   !        INPUT MASTER DATA
7620   PRINT PAGE,LIN(20)
7630   PRINT "INPUT MASTER DATA"
7640   INPUT "MASTER LATITUDE? D,M,S",D,M,S
7650   Xmit(4,1)=D+M/60+S/3600
7660   INPUT "MASTER LONGITUDE? D,M,S",D,M,S
7670   Xmit(4,2)=-(D+M/60+S/3600)
7680   INPUT "MASTER POWER LEVEL? KW",Power(4)
7690   !        INPUT SECONDARY DATA
7700   IF Conf=1 THEN L1$="TDX"
7710   IF Conf<>1 THEN L1$="TDW"
7720   IF (Conf=2) OR (Conf=3) THEN L2$="TDX"
7730   IF (Conf=1) OR (Conf=4) THEN L2$="TDY"
7740   IF (Conf=2) OR (Conf=3) THEN L3$="TDY"
7750   IF (Conf=1) OR (Conf=4) THEN L3$="TDZ"
7760   FOR I=1 TO 3
7770   PRINT PAGE,LIN(20)
7780  IF I=1 THEN PRINT "INPUT "&L1$&" DATA"
7790  IF I=2 THEN PRINT "INPUT "&L2$&" DATA"
7800  IF I=3 THEN PRINT "INPUT "&L3$&" DATA"
7810  INPUT "LATITUDE? D,M,S",D,M,S
7820  Xmit(I,1)=D+M/60+S/3600
7830  INPUT "LONGITUDE? D,M,S",D,M,S
7840  Xmit(I,2)=-(D+M/60+S/3600)
7850  INPUT "POWER LEVEL? KW",Power(I)
7860  INPUT "EMISSION DELAY?",Emis(I)
7870  NEXT I
7880  !           PRINT DATA TABLE AND CHECK FOR ERRORS
7890  PRINT PAGE
7900  PRINT "STATION";TAB(15);"LAT";TAB(30);"LON";TAB(45);"POWER";TAB(60);"EMIS
SION DELAY"
7910  PRINT LIN(2)
7920 ! Ld=INT(Xmit(4,1))
7930 ! Lm=INT(FRACT(Xmit(4,1))*60)
7940 ! Ls=FRACT(FRACT(Xmit(4,1))*60)*60
7950 ! PRINT Ld,Lm,Ls
7960  FIXED 4
7970  PRINT "MASTER";TAB(15);Xmit(4,1);TAB(30);Xmit(4,2);TAB(45);Power(4)
7980  PRINT L1$;TAB(15);Xmit(1,1);TAB(30);Xmit(1,2);TAB(45);Power(1);TAB(60);Em
is(1)
7990  PRINT L2$;TAB(15);Xmit(2,1);TAB(30);Xmit(2,2);TAB(45);Power(2);TAB(60);Em
is(2)
```

```
8000   PRINT L3$;TAB(15);Xmit(3,1);TAB(30);Xmit(3,2);TAB(45);Power(3);TAB(60);Em
is(3)
8010   INPUT "IS DATA CORRECT? Y OR N",C$
8020   !           IF DATA INCORRECT REENTER
8030   IF C$="N" THEN 7610
8040   !           FILE DATA
8050   CREATE Chain$,1
8060   ASSIGN #1 TO Chain$
8070   PRINT #1;Xmit(*),Power(*),Emis(*)
8080    GOTO Menu
8100    !
```

```
8110 K27: !  ^    PROJECT TDs

8120  PRINTER IS 16
8130  MASS STORAGE IS ":T15"
8140  PRINT PAGE
8150  PRINT "        MEASURED-PROJECTED TDs           "
8160  !              INPUT CHAIN DATA
8170  IF C1=1 THEN 8250
8190  INPUT "CHAIN FILE?",Chain$
8200  C1=1
8220  ASSIGN #1 TO Chain$
8230  READ #1;Xmit(*),Power(*),Emis(*)
8250  Conf=VAL(Chain$[5,5])
8260  Ch$=Chain$[1,4]
8270  !              INPUT WAYPOINT FILE DATA
8280  IF F1=1 THEN 8330
8290  INPUT "WAYPOINT FILE NAME?",File$
8300  F1=1
8310  ASSIGN #1 TO File$
8320  READ #1;Wpt(*)
8330  !              CALCULATE XY COORDINATES OF XMITTERS
8340  CALL Cart_coord(Xmit(*),Wpt(25,7),Wpt(25,8),Zxmit(*))
8350  !              INPUT WAYPOINT OF INTEREST
8360  INPUT "WAYPOINT FROM WHERE DATA IS TO BE PROJECTED?",W
8370  !              CALCULATE REFLECTED TD ARRAYS
8380  CALL Project(W,Wpt(*),W(*),X(*),Y(*),Z(*),Pos_x(*),Pos_y(*),Conf,Zxmit(*),
N,V)
8390  !              CALCULATE STATS AND PRINT RESULTS
8400  PRINTER IS 0
8410  PRINT "RESULTS OF CALCULATING MEASURED TDs-PROJECTED TDs FROM WP";W;"; FIL
E=";F$
8420  F$=F$&">"
8430  PRINTER IS 16
8440  GOTO K2   ! CALCULATE STATS
8460  !
```

```
8480  MASS STORAGE IS ":T15"
8490  PRINTER IS 16
8500  PRINT PAGE
8510  PRINT "      SIMULATE WAYPOINT SURVEY      "
8520  !       INPUT CHAIN DATA
8530  IF Ci=1 THEN 8610
8550  INPUT "CHAIN FILE?",Chain$
8560  Ci=1
8580  ASSIGN #1 TO Chain$
8590  READ #1;Xmit(*),Power(*),Emis(*)
8610  Conf=VAL(Chain$[5,5])
8620  Ch$=Chain$[1,4]
8630  !          INPUT WAYPOINT FILE DATA
8640  IF Fi=1 THEN 8690
8650  INPUT "WAYPOINT FILE NAME?",File$
8660  Fi=1
8670  ASSIGN #1 TO File$
8680  READ #1;Wpt(*)
8690  !          INPUT WAYPOINT
8700  PRINTER IS 0
8710  INPUT "WAYPOINT OF INTEREST?",Wp
8720  PRINT "WAYPOINT=";Wp
8730  !          CALCULATE ANGLES TO AJOINING WAYPOINTS
8740  CALL Bear(Wpt(Wp-1,7),Wpt(Wp-1,8),Wpt(Wp,7),Wpt(Wp,8),Ca)
8750  CALL Bear(Wpt(Wp+1,7),Wpt(Wp+1,8),Wpt(Wp,7),Wpt(Wp,8),Cb)
8760  FIXED 1
8770  PRINT "COURSE LINE ONE=";Ca;TAB(45);"COURSE LINE TWO=";Cb
8780  !          INPUT START AND STOP POINTS
8790  INPUT "START POINT RELATIVE TO WAYPOINT(KM)",Ds
8800  INPUT "STOP POINT RELATIVE TO WAYPOINT(KM)",De
8810  PRINT "START POINT=";Ds;" KM FROM WP";Wp;TAB(45);"STOP POINT=";De;" KM FR
M WP";Wp
8820  !          INPUT NOISE LEVEL AND NUMBER OF SAMPLES
8830  INPUT "EXPECTED STANDARD DEVIATION OF TDS(MICROSEC)",Sigma
8840  INPUT "NUMBER OF SAMPLES PER TRACKLINE",N
8850  FIXED 3
8860  PRINT "NOISE=";Sigma;" MICROSEC";TAB(45);"SAMPLES=";N
8870  !          CALCULATE AND REDUCE TRACKLINE ONE DATA
8880  Zp(1)=Wpt(Wp,7)
8890  Zp(2)=Wpt(Wp,8)
8900  CALL Data(W(*),X(*),Y(*),Z(*),N,Ca,Ds,De,Sigma,Xmit(*),Zp(*),V,Conf)
8910  CALL Stat_mat(W(*),X(*),Y(*),Z(*),Stat(*),Cov(*),O_set(*),N)
8920  CALL Reg(Stat(*),Cov(*),S1(*),Rone(*),O_set(*))
8930  PRINT "TRACKLINE ONE: WAYPOINT ";Wp-1;" TO WAYPOINT ";Wp
8940  CALL Print(Stat(*),O_set(*),Cov(*),Rone(*),N,1)
8950  !          CALCULATE AND REDUCE TRACKLINE TWO DATA
8960  CALL Data(W(*),X(*),Y(*),Z(*),N,Cb,Ds,De,Sigma,Xmit(*),Zp(*),V,Conf)
8970  CALL Stat_mat(W(*),X(*),Y(*),Z(*),Stat(*),Cov(*),O_set(*),N)
8980  CALL Reg(Stat(*),Cov(*),S2(*),Rtwo(*),O_set(*))
8990  PRINT "TRACKLINE TWO: WAYPOINT ";Wp+1;" TO WAYPOINT ";Wp
9000  CALL Print(Stat(*),O_set(*),Cov(*),Rtwo(*),N,1)
9010  !          CALCULATE WAYPOINT
9020  CALL Way(S1(*),S2(*),Rone(*),Rtwo(*),N,N,Wp(*))
```

```
9030  CALL Wprint(Wp(*))
9040  GOTO Menu
9050 END
9051  !
```

```
9060 SUB Reg(Stat(*),Cov(*),S(*),R(*),Offset(*))
9070 OPTION BASE 1
9080 DEFAULT ON
9090 S(1,1)=S(1,2)=S(1,3)=Stat(3,1)+Offset(1)
9100 S(1,4)=S(1,5)=Stat(3,2)+Offset(2)
9110 S(1,6)=Stat(3,3)+Offset(3)
9120 S(2,1)=Stat(3,2)+Offset(2)
9130 S(2,2)=S(2,4)=Stat(3,3)+Offset(3)
9140 S(2,3)=S(2,5)=S(2,6)=Stat(3,4)+Offset(4)
9150 S(3,1)=S(3,2)=S(3,3)=Stat(4,1)
9160 S(3,4)=S(3,5)=Stat(4,2)
9170 S(3,6)=Stat(4,3)
9180 S(4,1)=Stat(4,2)
9190 S(4,2)=S(4,4)=Stat(4,3)
9200 S(4,3)=S(4,5)=S(4,6)=Stat(4,4)
9210 FOR I=1 TO 6
9220 Ind_var:!
9230 Iv=1
9240 D=2
9250 IF S(4,I)>S(3,I) THEN Iv=2
9260 IF S(4,I)>S(3,I) THEN D=1
9270 R(3,I)=Iv
9280 Slope:!
9290 R(1,I)=Cov(3,I)*S(D+2,I)/S(Iv+2,I)
9300 IF Iv=2 THEN R(1,I)=1/R(1,I)
9310 Residual:!
9320 C=1-Cov(3,I)^2
9330 IF C<0 THEN 9350
9340 R(2,I)=S(D+2,I)*C^.5
9350 NEXT I
9360 SUBEND
9370  !
```

```
9390 SUB Print(Stats(*),O_set(*),Cov(*),R(*),Sample,V)
9400  OPTION BASE 1
9410  FOR I=1 TO 3
9420  FOR J=1 TO 6
9430  IF R(I,J)>100 THEN R(I,J)=0
9440  IF Cov(3,J)>100 THEN Cov(3,J)=0
9450  NEXT J
9460  NEXT I
9470   PRINT RPT$("_",80)
9480   IF V=2 THEN 9630
9490   PRINT LIN(1);TAB(21);" TDW ";TAB(36);" TDX ";TAB(51);" TDY ";TAB(66);" TD
Z "
9500   FIXED 3
9510   PRINT LIN(1);TAB(1);"CUMULATIVE AVERAGE";TAB(20);Stats(3,1)+O_set(1);TAB(
35);Stats(3,2)+O_set(2);TAB(50);Stats(3,3)+O_set(3);TAB(65);Stats(3,4)+O_set(4)
9520   PRINT LIN(1);TAB(1);"STANDARD DEVIATION";TAB(20);Stats(4,1);TAB(35);Stats
(4,2);TAB(50);Stats(4,3);TAB(65);Stats(4,4)
9530   PRINT LIN(1);TAB(1);"TD PAIR";TAB(17);"WX";TAB(27);"WY";TAB(37);"WZ";TAB(
47);"XY";TAB(57);"XZ";TAB(67);"YZ"
9540   PRINT LIN(1);TAB(1);"CORR COEF";TAB(15);Cov(3,1);TAB(25);Cov(3,2);TAB(35)
;Cov(3,3);TAB(45);Cov(3,4);TAB(55);Cov(3,5);TAB(65);Cov(3,6)
9550   PRINT LIN(1);TAB(1);"SLOPE";TAB(15);R(1,1);TAB(25);R(1,2);TAB(35);R(1,3);
TAB(45);R(1,4);TAB(55);R(1,5);TAB(65);R(1,6)
9560   PRINT LIN(1);TAB(1);"RESIDUAL";TAB(15);R(2,1);TAB(25);R(2,2);TAB(35);R(2,
3);TAB(45);R(2,4);TAB(55);R(2,5);TAB(65);R(2,6)
9570   STANDARD
9580   PRINT LIN(1);TAB(1);"IND VAR";TAB(17);R(3,1);TAB(27);R(3,2);TAB(37);R(3,3
);TAB(47);R(3,4);TAB(57);R(3,5);TAB(67);R(3,6)
9590   PRINT LIN(1);"SAMPLES=";Sample
9600   PRINT RPT$("_",80)
9610   PRINT LIN(1)
9620   SUBEXIT
9630   FIXED 3
9640   PRINT LIN(1);TAB(21);" R1  ";TAB(36);" R2  ";TAB(51);" X   ";TAB(66);" Y
 "
9650   PRINT LIN(1);TAB(1);"CUMULATIVE AVERAGE";TAB(20);Stats(3,1)+O_set(1);TAB(
35);Stats(3,2)+O_set(2);TAB(50);Stats(3,3)+O_set(3);TAB(65);Stats(3,4)+O_set(4)
9660   PRINT LIN(1);TAB(1);"STANDARD DEVIATION";TAB(20);Stats(4,1);TAB(35);Stats
(4,2);TAB(50);Stats(4,3);TAB(65);Stats(4,4)
9670   PRINT RPT$("_",80)
9680   SUBEND
9690   !
```

```
9710 SUB Hi_lo(X(*),N,Hi,Lo)
9720 Hi=X(1)
9730 Lo=X(1)
9740 FOR I=1 TO N
9750 Lo=MIN(X(I),Lo)
9760 Hi=MAX(X(I),Hi)
9770 NEXT I
9780 SUBEND
```

```
9790 SUB Plot(Lx,Hx,Ly,Hy,Xx,Yx,X(*),Y(*),S1,N,X$,Y$,F$)
9800 OPTION BASE 1
9810 PLOTTER IS 13,"GRAPHICS"
9820 GRAPHICS
9830 LOCATE 10,90,20,100
9840 SHOW Lx-1,Hx+1,Ly-1,Hy+1
9850 AXES 1,1,Xx,Yx
9860 LINE TYPE 2
9870 FOR I=1 TO N
9880 DRAW X(I),Y(I)
9890 NEXT I
9900 LINE TYPE 1
9910 MOVE Lx,S1*(Lx-Xx)+Yx
9920 DRAW Hx,S1*(Hx-Xx)+Yx
9930 Label:CSIZE 3
9940  LORG 5
9950  LOCATE 0,125,0,20
9960  SCALE 0,100,0,20
9970  MOVE 50,15
9980   LABEL USING "#,K";X$;" VS ";Y$;" & REGRESSION LINE"
9990  MOVE 50,10
10000 LABEL USING "K";"ONE MICROSEC/DIV"
10010 MOVE 50,5
10020 LABEL USING "K";"FILE="&F$
10030 PAUSE
10040 INPUT "HARD COPY? Y OR N",Hc$
10050 IF Hc$<>"Y" THEN 10070
10060 DUMP GRAPHICS
10070 EXIT GRAPHICS
10080 SUBEND
10100 !
```

```
10110 SUB Rplot(Iv(*),D(*),S(*),R(*),Pr,N,V1$,V2$,F$)
10120 OPTION BASE 1
10130 P1=1
10140 PRINTER IS 0
10150 INPUT "PLOT RESIDUALS VS N(1) OR INDEPENDENT VAR(2)",P1
10160 PLOTTER IS 13,"GRAPHICS"
10170 GRAPHICS
10180 LOCATE 10,100,20,80
10190 IF P1=1 THEN 10240
10200 CALL Hi_lo(Iv(*),N,Imax,Imin)
10210 SCALE Imin-2,Imax,-5,5
10220 AXES 1,1,Imin,0
10230 GOTO 10260
10240 SCALE 0,N,-5,5
10250 AXES 10,1,0,0
10260 LINE TYPE 2
10270 R$=VAL$(R(2,Pr))
10280 M=R(1,Pr)
10290 IF R(3,Pr)=2 THEN M=1/M
10300 Iv=1
10310 D=2
10320 IF R(3,Pr)=2 THEN Iv=2
10330 IF R(3,Pr)=2 THEN D=1
10340 Da=S(D,Pr)
10350 Ia=S(Iv,Pr)
10360 FOR I=1 TO N
10370 Dp=Da+M*(Iv(I)-Ia)
10380 Dr=Dp-D(I)
10390 Nr=Dr/R(2,Pr)
10400 IF Nr>5 THEN PRINT I,Nr
10410  IF P1=1 THEN 10440
10420  DRAW Iv(I),Nr
10430  GOTO 10450
10440 DRAW I,Nr
10450 NEXT I
10460 PRINTER IS 16
10470 LINE TYPE 1
10480 IF P1=1 THEN 10610
10490  Ra=Imax-Imin
10500  MOVE Imin+Ra/8,-4
10510 LABEL USING "K";"NORMALIZED RESIDUALS VS INDEP VAR"
10520 MOVE Imin+Ra/8,-5
10530 LABEL USING "K";V1$;" AND ";V2$;" DATA";"; RES=";R$;" MICROSEC"
10540 MOVE Imin+Ra/8,-6
10550 LABEL USING "K";"TRACKLINE=";F$
10560 MOVE Imin+Ra/8,4
10570 LABEL USING "K";Imin
10580 MOVE Imin+Ra/8,5
10590 LABEL USING "K";"IND VAR MIN"
10600 GOTO 10670
10610 MOVE N/8,-4
10620 LABEL USING "K";"NORMALIZED RESIDUALS VS SAMPLE NUMBER"
10630 MOVE N/8,-5
10640 LABEL USING "K";V1$;" AND ";V2$;" DATA";"; RES=";R$;" MICROSEC"
10650 MOVE N/8,-6
```

```
10660 LABEL USING "K";"TRACKLINE=";F$
10670 INTEGER K
10680 IF Pi=1 THEN 10710
10690 K=Imax-Imin
10700 GOTO 10750
10710 K=N/10
10720 IF (N>100) AND (Pi=1) THEN K=N/50
10730 Step=1
10740 IF (Pi=2) AND (K>10) THEN Step=2
10750 FOR I=1 TO K
10760 Mx=Imin+I
10770 IF Pi=1 THEN Mx=10*I
10780 IF (N>100) AND (Pi=1) THEN Mx=50*I
10790 MOVE Mx,-1
10800 IF Pi=1 THEN L=Mx
10810 IF Pi=2 THEN L=I
10820 LABEL USING "K";L
10830 NEXT I
10840 IF Pi=2 THEN 10930
10850 FOR I=1 TO 5
10860 M=N
10870 MOVE -M/16,I
10880 LABEL USING "K";I
10890 MOVE -M/16,-I
10900 LABEL USING "K";-I
10910 NEXT I
10920 GOTO 10990
10930 FOR I=1 TO 5
10940 MOVE Imin-1,I
10950 LABEL USING "K";I
10960 MOVE Imin-1,-I
10970 LABEL USING "K";-I
10980 NEXT I
10990 PAUSE
11000 INPUT "HARD COPY? Y OR N",Hc$
11010 IF Hc$="Y" THEN DUMP GRAPHICS
11020 EXIT GRAPHICS
11030 SUBEND
11050 !
```

```
11060 SUB Bear(P1,P2,P3,P4,B)
11070 ! PROGRAM TO CALCULATE BEARING BETWEEN TWO POINTS
11080 ! (P1,P2) AND (P3,P4).POSITIONS ARE INPUT IN DEGREES
11090 ! OF LAT/LON . N AND E ARE +; S AND W ARE -
11100 DEFAULT ON
11110 DEG
11120 P5=COS(P3)
11130 P6=P4-P2
11140 P7=SIN(P3)*COS(P1)-P5*SIN(P1)*COS(P6)
11150 B=ATN(P5*SIN(P6)/P7)
11160 IF P7<0 THEN B=B+180
11170 IF B<0 THEN B=B+360
11180 SUBEND
11210 !
```

```
11220 SUB Range(P1,P2,P3,P4,R)
11230 ! PROGRAM TO CALCULATE RANGE IN KILOMETERS BETWEEN TWO
11240 ! POINTS (P1,P2) AND (P3,P4). POSITIONS ARE IN DEGREES
11250 ! (LAT,LON) : NORTH AND EAST ARE +.
11260 DEG
11270 P6=SIN(P1)
11280 P7=SIN(P3)
11290 P8=P6*P7+COS(P1)*COS(P3)*COS(P4-P2)
11300 RAD
11310 P9=(1-P8^2)^.5
11320 P5=ATN(P9/P8)
11330 P10=(P6+P7)^2*(3*P9-P5)/(1+P8)
11340 P11=(P6-P7)^2*(3*P9+P5)/(1-P8)
11350 DEG
11360 R=6378.135*P5+5.346125*(P10-P11)
11370 SUBEND
11400 !
```

```
11410 SUB Xy(Xp,Yp,R,B,X,Y)
11420 ! PROGRAM CALCULATES XY COORDINATES OF POINT(Xp,Yp)
11430 ! FROM RANGE AND BEARING FROM REFERENCE POINT(X,Y)
11440 DEG
11450 Xp=X+R*SIN(B)
11460 Yp=Y+R*COS(B)
11470 SUBEND
11500 !
```

```
11510 SUB G_mat(P(*),V,Range(*),Bear(*),Zp(*),A(*),G123(*),G12(*),G23(*),G13(*))
11520 ! PROGRAM CALCULATES:
11530 !          A(*);GRADIENT MATRIX(XY TO TD)
11540 !          G123(*);GRADIENT MATRIX (TD TO XY),3-TD CASE
11550 !          G12(*),G23(*),G13(*);GRADIENT MATRICES(TD TO XY),2-TD CASES
11560 !        INPUTS ARE:
11570 !          P(*);POWER LEVEL OF XMITTERS (KILOWATTS)
11580 !          V;VELOCITY OF PROPOGATION (MICROSEC/KILOMETER)
11590 !          Range(*);RANGES FROM POINT TO XMITTERS (KILOMETERS)
11600 !          Bear(*);AZIMUTH OF XMITTERS FROM POINT (DEGREES)
11610 !          Zp(*);XY POSITION OF POINT
11620 DEG
11630 OPTION BASE 1
11640 DIM R(3,3),W(3,3),B(2,3),C(2,2),D(2,3)
11650 MAT R=(1)
11660 K=P(4)/Range(4)^2
11670 FOR I=1 TO 3
11680 ! CALCULATE A MATRIX
11690 A(I,1)=(SIN(Bear(4))-SIN(Bear(I)))/V
11700 A(I,2)=(COS(Bear(4))-COS(Bear(I)))/V
11710 ! CALCULATE COVARIANCE MATRIX
11720 R(I,I)=1+K*Range(I)^2/P(I)
11730 NEXT I
11740 ! CALCULATE WEIGHTING MATRIX
11750 MAT W=INV(R)
11760 ! CALCULATE G123
11770 MAT B=TRN(A)
11780 MAT D=B*W
11790 MAT C=D*A
11800 MAT C=INV(C)
11810 MAT G123=C*D
11820 ! CALCULATE G12,G23,G13
11830 FOR I=1 TO 2
11840 FOR J=1 TO 2
11850 K=I+1
11860 L=J+1
11870 M=I
11880 N=J
11890 IF I=2 THEN M=3
11900 IF J=2 THEN N=3
11910 G12(I,J)=A(I,J)
11920 G23(I,J)=A(K,J)
11930 G13(I,J)=A(M,J)
11940 NEXT J
11950 NEXT I
11960 MAT G12=INV(G12)
11970 MAT G23=INV(G23)
11980 MAT G13=INV(G13)
11990 SUBEND
12020 !
```

```
12030 SUB Td(R(*),V,Ed(*),T(*))
12040 ! PROGRAM PREDICTS TDS FOR A POINT. INPUTS ARE RANGES
12050 ! TO THE XMITTERS(R(*)),EMISSION DELAYS(Ed(*)),AND
12060 ! VELOCITY OF PROPOGATION
12070 !
12080 ! CALCULATE SECONDARY PHASE FACTORS (SF)
12090 FOR I=1 TO 4
12110 S(I)=38.4488/R(I)-.40758+.002166735*R(I)
12130 IF R(I)<160 THEN S(I)=.816768/R(I)-.011402+.0011*R(I)
12140 NEXT I
12150 ! CALCULATE TDS
12160 FOR I=1 TO 3
12170 T(I)=(R(I)-R(4))/V+S(I)-S(4)+Ed(I)
12180 NEXT I
12190 SUBEND
12191 !
```

```
12200  SUB Wpfile(W(*),Fi,File$)
12210 ! PROGRAM TO STORE WAYPOINT INFO
12220 OPTION BASE 1
12230 PRINTER IS 0
12240 IF Fi=1 THEN Opt
12250 INPUT "WAYPOINT FILE NAME?",File$
12260 Fi=1
12270 INPUT "DOES THIS FILE CURRENTLY EXIST ON TAPE? Y OR N",Cr$
12280 IF UPC$(Cr$[1,1])="N" THEN CREATE File$,8
12300 Cont: ASSIGN #1 TO File$
12310 IF UPC$(Cr$[1,1])="Y" THEN READ #1;W(*)
12311 Opt:  INPUT "MODIFY/CREATE 1; LIST 2; SAVE 3; QUIT 4",Opt
12312 IF (Opt<1) OR (Opt>4) THEN Opt
12313 ON Opt GOTO Input,List,File,End
12330 Input: INPUT "WAYPOINT NUMBER?",Nw
12340 PRINT "WAYPOINT NUMBER=";Nw
12350 INPUT "INPUT OR CHANGE TDs? Y OR N",Td$
12360 IF UPC$(Td$)<>"Y" THEN Xy
12370 INPUT "TDW,TDX,TDY,TDZ ?",W(Nw,1),W(Nw,2),W(Nw,3),W(Nw,4)
12380 FIXED 2
12390 PRINT "TDW=";W(Nw,1),"TDX=";W(Nw,2),"TDY=";W(Nw,3),"TDZ=";W(Nw,4)
12400 Xy:INPUT "INPUT OR CHANGE XY POSITION? Y OR N",P$
12410 IF UPC$(P$)<>"Y" THEN L1
12420 FIXED 3
12430 INPUT "XY POSITION? E,N",W(Nw,5),W(Nw,6)
12440 PRINT "X=";W(Nw,5),"Y=";W(Nw,6)
12450 L1:INPUT "INPUT OR CHANGE LAT/LON? Y OR N",L1$
12460 IF UPC$(L1$)<>"Y" THEN Next
12470 FIXED 4
12480 INPUT "ENTER LATITUDE; D,M,S",Deg,Min,Sec
12481 W(Nw,7)=Deg+Min/60+Sec/3600
12482 INPUT "ENTER LONGITUDE; D,M,S",Deg,Min,Sec
12483 W(Nw,8)=-(Deg+Min/60+Sec/3600)
12490 PRINT "LAT=";W(Nw,7),"LON=";W(Nw,8)
12500 Next:INPUT "ANOTHER WAYPOINT? Y OR N",A$
12510 IF A$="Y" THEN Input
12520 GOTO Opt
12521 List: PRINT LIN(5);TAB(25);"WAYPOINT FILE:  ";File$
12540 FOR I=1 TO 25
12550 FIXED 0
12560 PRINT
12570 PRINT "WAYPOINT=";I
12580 FIXED 2
12590 PRINT "TDW=";W(I,1),"TDX=";W(I,2),"TDY=";W(I,3),"TDZ=";W(I,4)
12600 FIXED 3
12610 PRINT "X=";W(I,5),"Y=";W(I,6)
12620 FIXED 0
12621 Lat=W(I,7)
12622 Deg=INT(Lat)
12623 Mini=60*(Lat-Deg)
12624 Min=INT(Mini)
12625 Sec=60*(Mini-Min)
12626 PRINT "LAT:";Deg;CHR$(179);Min;"'";
12627 FIXED 2
12628 PRINT Sec;"''    ";
```

```
12629 FIXED 0
12630 Lon=-W(I,8)
12631 Deg=INT(Lon)
12632 Mini=60*(Lon-Deg)
12633 Min=INT(Mini)
12634 Sec=60*(Mini-Min)
12635 PRINT "LON:";Deg;CHR$(179);Min;"'";
12636 FIXED 2
12637 PRINT Sec;"''"
12640 NEXT I
12641 PRINT LIN(5)
12642 GOTO Opt
12650 File:ASSIGN #1 TO File$
12680 PRINT #1;W(*)
12690 End:SUBEND
12710 !
```

```
12720 SUB Fehg(X(*),Zp(*),Tp(*),Tq(*),G(*),Zq(*),V,Nsta)
12730 ! THIS SUBROUTINE CALCULATES POSITION COORDINATES,Zq(2,1),
12740 ! USING THE FLAT EARTH HYPERBOLIC GRID (FEGH) ALGROTHIM.
12750 ! INPUTS ARE TRANSMITTER POSITIONS,X(2,3),WAYPOINT POSITION,
12760 ! Zp(2,1); WAYPOINT TDs, Tp(2,1); THE GRADIENT MATRIX, G(2,2) OR G(2,3);
12770 ! THE VELOCITY OF PROPOGATION, V; AND THE MEASURED TDs, Tq(2,1).
12780 OPTION BASE 1
12790 DIM A(Nsta,1),B(Nsta,1),C(Nsta,1),D(Nsta,1),E(2,1),T(Nsta,1),Z(2,1),R(Nsta
+1,1)
12800 Initialize:!
12810 K=0
12820 MAT Zq=Zp
12830 MAT B=Zp
12840 MAT A=Zp
12850 MAT C=Tq
12860 MAT D=Tp
12870 ! CALCULATE Zq(1)=Zp+G(Tq-Tp)
12880 MAT T=C-D
12890 MAT Zq=G*T
12900 MAT Zq=Zp+Zq
12910 ! CALULATE h(Zq(0))=h(Zp)
12920 FOR I=1 TO Nsta+1
12930 R(I,1)=SQR((Zp(1,1)-X(I,1))^2+(Zp(2,1)-X(I,2))^2)
12940 NEXT I
12950 FOR I=1 TO Nsta
12960 D(I,1)=(R(I,1)-R(Nsta+1,1))/V
12970 NEXT I
12980 ! CALCULATE Zq(n)=2Zq(n-1)-Zq(n-2)+G(h(Zq(n-2)-h(Zq(n-1)))
12990 Iterate:K=K+1
13000 DISP K
13010 ! IF ITERATIONS EXCEED 20 THEN STOP
13020 IF K>20 THEN Error
13030 MAT B=A ! Zq(n-2)=Zq(n-1)
13040 MAT A=Zq! Zq(n-1)=Zq(n)
13050 MAT C=D ! h(Zq(n-2))=h(Zq(n-1))
13060 ! CALCULATE h(Zq(n-1))
13070 FOR I=1 TO Nsta+1
13080 R(I,1)=SQR((Zq(1,1)-X(I,1))^2+(Zq(2,1)-X(I,2))^2)
13090 NEXT I
13100 FOR I=1 TO Nsta
13110 D(I,1)=(R(I,1)-R(Nsta+1,1))/V
13120 NEXT I
13130 MAT Zq=Zq*(2)
13140 MAT Zq=Zq-B
13150 MAT T=C-D
13160 MAT Z=G*T
13170 MAT Zq=Zq+Z
13180 ! CALCULATE ABS(Zq(n)-Zq(n-1))
13190 MAT E=A-Zq
13200 Diff=E(1,1)^2+E(2,1)^2
13210 ! IF DIFFERENCE >1 METER, THEN ITERATE
13220 IF SQR(Diff)>.0010 THEN Iterate
13230 DISP K
13240 GOTO End
13250 Error:DISP "20 ITERATIONS WITHOUT CLOSURE !!!!!"
```

```
13260 PAUSE
13270 End:SUBEND
13290 !
```

```
13300 SUB Xplot(T,Ax(*),Ay(*),Minx,Maxx,Miny,Maxy,N,No,F$,Ch$,L$,P1)
13310 DIM L1$[80],L4$[80],L2$[80],L3$[80]
13320 FIXED 3
13330 Zoom$=""
13331 Plotter=1
13340 IF No=2 THEN 13430
13341 INPUT "PLOT DATA ON CRT1 OR 9872A2? 1 OR 2",Plotter
13342 PLOTTER IS 6,5,"9872A"
13350 IF Plotter=1 THEN PLOTTER IS 13,"GRAPHICS"
13351 IF P1<>5 THEN 13370
13352 INPUT "CHART SCALE  1:10,0001, 1:20,0002, 1:40,0004, OR 1:80,0008?
 1,2,4,OR8",Scale
13353 IF (Scale<>1) AND (Scale<>2) AND (Scale<>4) AND (Scale<>8) THEN 13352
13354 Mult=100/Scale
13356 Xo=Minx*Mult
13357 Yo=Miny*Mult
13358 X=Maxx*Mult
13359 Y=Maxy*Mult
13360 ! GRAPHICS
13361 IF P1=5 THEN MSCALE X,Y
13370 IF P1<>5 THEN LOCATE 0,100,20,100
13371 GRAPHICS
13380 IF P1<>5 THEN FRAME
13390  IF (P1<>5) AND (Zoom$<>"Y") THEN SHOW Minx-.1,Maxx+.1,Miny-.1,Maxy+.1
13400  IF (P1<>5) AND (Zoom$="Y") THEN SHOW Minx,Maxx,Miny,Maxy
13410 IF P1<>5 THEN AXES .1,.1,Minx,Miny,10,10
13411 IF P1=5 THEN AXES .1*Mult,.1*Mult,0,0,10,10
13420 ! GRID 2,2,Minx,Miny,1,1
13430 LINE TYPE T
13440 IF P1<>5 THEN MOVE Ax(1),Ay(1)
13441 IF P1=5 THEN MOVE Ax(1)*Mult-Xo,Ay(1)*Mult-Yo
13450 FOR I=1 TO N
13460 IF P1<>5 THEN DRAW Ax(I),Ay(I)
13461 IF P1=5 THEN DRAW Ax(I)*Mult-Xo,Ay(I)*Mult-Yo
13470 NEXT I
13471 IF (P1=5) AND (Plotter=1) THEN PAUSE
13472 IF (P1=5) AND (Plotter=1) THEN 14080
13473 IF (P1=5) AND (Plotter=2) THEN 14100
13480 PRINTER IS 0
13490 P=0
13500 IF (P1=3) AND (No=1) THEN SUBEXIT
13510 IF (P1=3) AND (No=2) THEN Label
13520 PAUSE
13530 INPUT "ZOOM?,Y OR N",Zoom$
13540 IF Zoom$<>"Y" THEN 13580
13550 DIGITIZE Minx,Miny
13560 DIGITIZE Maxx,Maxy
13570 GOTO 13350
13580 INPUT "FIND SAMPLE NUMBER AND VALUE OF PLOTTED DATA POINT? Y OR N",Out$
13590 IF Out$<>"Y" THEN 13690
13600 DIGITIZE X,Y
13610 Tol=.1
13620 IF ABS(Maxx-Minx)<=1 THEN Tol=.010
13630 FOR I=1 TO N
13640 IF (ABS(Ax(I)-X)<Tol) AND (ABS(Ay(I)-Y)<Tol) THEN P=1
```

```
13650 IF P THEN PRINT I,Ax(I),Ay(I)
13660 P=0
13670 NEXT I
13680 GOTO 13580
13690 INPUT "DIGITIZE?",D$
13700 IF D$<>"Y" THEN Label
13710 INPUT "HOW MANY?<=10",P
13720 IF P>10 THEN P=10
13730 GRAPHICS
13740 FOR I=1 TO P
13750 DIGITIZE Xx(I),Yy(I)
13760 LORG 5
13770 MOVE Xx(I),Yy(I)
13780 LABEL USING "K";"+"
13790 MOVE Xx(I),Yy(I)
13800 LORG 2
13810 LABEL USING "3D";I
13820 NEXT I
13830 FOR I=1 TO P
13840 PRINT I,Xx(I),Yy(I)
13850 NEXT I
13860 Label:IF (Pl=3) AND (No=1) THEN SUBEXIT
13870 LOCATE 0,100,0,15
13880 SCALE 0,100,0,25
13890 L1$="AXES: X="&VAL$(Minx)&"  Y="&VAL$(Miny)
13900 L2$="UNITS: 1 KM/DIV"
13910 L3$="FILE="&F$
13920 IF Pl=1 THEN Mini
13930 IF Pl=3 THEN Mini_lc
13940 L4$="LORAN-C DATA:"&"  CHAIN="&Ch$&", LOPs="&L$
13950 GOTO 13990
13960 Mini:L4$="MINI-RANGER DATA"
13970 GOTO 13990
13980 Mini_lc: L4$="MINI-RANGER (SOLID LINE) AND LORAN-C (DASHED LINE)"
13990 MOVE 10,20
14000 LABEL L1$
14010 MOVE 10,15
14020 LABEL L2$
14030 MOVE 10,10
14040 LABEL L3$
14050 MOVE 10,5
14060 LABEL L4$
14070 PAUSE
14080 Hrd_cop:INPUT "HARD COPY OF GRAPHICS?",H$
14090 IF H$="Y" THEN DUMP GRAPHICS
14100 EXIT GRAPHICS
14110 SUBEND
14130 !
```

```
14140 SUB Ct_at(X(*),Y(*),Ct(*),At(*),Wx1,Wy1,Wx2,Wy2,N,R,A)
14150 DEFAULT ON
14160 OPTION BASE 1
14170 DEG
14180 MAT Ct=(0)
14190 MAT At=(0)
14200 ! CALCULATE COURSE ANGLE FROM W1 TO W2
14210 Dy=Wy2-Wy1
14220 Dx=Wx2-Wx1
14230 R=SQR(Dx^2+Dy^2)
14240 A=ATN(Dx/Dy)
14250 IF Dy<0 THEN A=A+180
14260 IF A<0 THEN A=A+360
14270 ! CALCULATE ALONG TRACK AND CROSS TRACK DISTANCE
14280 FOR I=1 TO N
14290 X=X(I)-Wx1
14300 Y=Y(I)-Wy1
14310 At(I)=X*SIN(A)+Y*COS(A)
14320 Ct(I)=-X*COS(A)+Y*SIN(A)
14330 NEXT I
14340 ! FILTER Ct
14350   FOR I=2 TO N
14360   Ct(I)=Ct(I-1)*.7+Ct(I)*.3
14370   NEXT I
14380 SUBEND
14400 !
```

```
14410 SUB Patct(At(*),Ct(*),N,Mina,Maxa,Minc,Maxc,Wt,Wf,F$,R,No,Data)
14420 DIM L1$[80],L2$[80],L3$[80],L4$[80]
14430 IF No=2 THEN 14540
14440 PLOTTER IS 13,"GRAPHICS"
14450 GRAPHICS
14460 LOCATE 10,100,20,100
14470 Mina=MIN(0,Mina)
14480 Maxa=MAX(R,Maxa)
14490 ! CALL Hi_lo(Ct(*),N,Bc,Lc)
14500 Bc=MAX(Maxc,.22)
14510 Lc=MIN(Minc,-.22)
14520 SCALE Mina,Maxa,Lc,Bc
14530 AXES .1,.01,0,0,10,10
14540 MOVE At(1),Ct(1)
14550 LINE TYPE 1
14560 IF (Data=3) AND (No=2) THEN LINE TYPE 2
14570 FOR I=1 TO N
14580 DRAW At(I),Ct(I)
14590 NEXT I
14600 MOVE R,0
14610 LORG 5
14620 LABEL USING "K";"I"
14630 LINE TYPE 1
14640 IF (Data=3) AND (No=1) THEN SUBEXIT
14650 Label:!
14660 Y_axis:!
14670 Y=.2
14680 IF Maxa>5 THEN Y=.5
14690 MOVE Y,.1
14700 LABEL USING "K";100
14710 MOVE Y,.2
14720 LABEL USING "K";200
14730 MOVE Y,-.1
14740 LABEL USING "K";-100
14750 MOVE Y,-.2
14760 LABEL USING "K";-200
14770 MOVE Y,-.215
14780 LABEL USING "K";"METERS"
14790 X_axis:!
14800 INTEGER K
14810 K=Maxa
14820 FOR I=1 TO K
14830 MOVE I,-.015
14840 LABEL USING "K";I
14850 NEXT I
14860 MOVE K-1,-.035
14870 LABEL USING "K";"KILOMETERS"
14880 LOCATE 10,100,0,20
14890 SCALE 0,100,0,25
14900 FIXED 0
14910 L1$="ALONG TRACK VS CROSS TRACK "
14920 L2$="WAYPOINT "&VAL$(Wf)&" TO "&"WAYPOINT "&VAL$(Wt)
14930 L3$="FILE="&F$
14940 IF Data=1 THEN L4$="MINIRANGER DATA"
14950 IF Data=2 THEN L4$="LORAN-C DATA"
```

```
14960 IF Data=3 THEN L4$="LORAN-C DATA(DOTTED LINE) AND MINIRANGER DATA(SOLID LI
NE)"
14970 MOVE 0,22
14980 LABEL "WP"&VAL$(Wt)
14990 MOVE 100,22
15000 LABEL "WP"&VAL$(Wf)
15010 MOVE 55,20
15020 LABEL L1$
15030 MOVE 55,15
15040 LABEL L2$
15050 MOVE 55,10
15060 LABEL L3$
15070 MOVE 55,5
15080 LABEL L4$
15090 PAUSE
15100 INPUT "HARD COPY? Y OR N",Hc$
15110 IF Hc$="Y" THEN DUMP GRAPHICS
15120 EXIT GRAPHICS
15130 SUBEND
15150 !
```

```
15160 SUB Delete(W(*),X(*),Y(*),Z(*),R1(*),R2(*),T$(*),N,F$)
15170 OPTION BASE 1
15180 DIM We(400),Xe(400),Ye(400),Ze(400),Te$(400),R1e(400),R2e(400)
15190 PRINTER IS 0
15200 Sample:INPUT "SAMPLE TO BE DELETED?START WITH HIGHEST NUMBER ",D
15210 J=0
15220 FOR I=1 TO N
15230 Sampl:IF I=D THEN Next
15240 J=J+1
15250 DISP I,J
15260 We(J)=W(I)
15270 Xe(J)=X(I)
15280 Ye(J)=Y(I)
15290 Ze(J)=Z(I)
15300 R1e(J)=R1(I)
15310 R2e(J)=R2(I)
15320 Te$(J)=T$(I)
15330 Next: NEXT I
15340 N=J
15350 MAT W=We
15360 MAT X=Xe
15370 MAT Y=Ye
15380 MAT Z=Ze
15390 MAT R1=R1e
15400 MAT R2=R2e
15410 FOR I=1 TO N
15420 T$(I)=Te$(I)
15430 NEXT I
15440 PRINT "SAMPLE";D;"DELETED,";N;"SAMPLES REMAIN"
15450 INPUT "ANOTHER SAMPLE TO DELETE?",A$
15460 IF A$="Y" THEN Sample
15540 End:SUBEND
15560 !
```

```
15570 SUB Read(W(*),X(*),Y(*),Z(*),R1(*),R2(*),Time$(*),Sample,F$)
15580 OPTION BASE 1
15590 DIM A(100),B(100),C(100),D(100),T$(100),E(100),F(100)
15600 MASS STORAGE IS ":T14"
15610 Sample=0                    ! ZERO SAMPLE NUMBER AND DATA ARRAYS
15620 MAT W=(0)
15630 MAT X=(0)
15640 MAT Y=(0)
15650 MAT Z=(0)
15660 MAT R1=(0)
15670 MAT R2=(0)
15680 FOR I=1 TO 400
15690 Time$(I)=""
15700 NEXT I
15710 INPUT "  FILE NAME? ",F$    ! INPUT FILE NAME
15720 ASSIGN #1 TO F$
15730   READ #1;Set,Sample        ! READ NUMBER OF DATA SETS AND SAMPLES
15740 FOR J=0 TO 3                ! READ DATA SET BY SET AND LOAD INTO
15750 ON END #1 GOTO 15770        ! DATA ARRAYS
15760 READ #1;A(*),B(*),C(*),D(*),E(*),F(*),T$(*)
15770 FOR I=1 TO 100
15780 W(I+J*100)=A(I)
15790 X(I+J*100)=B(I)
15800 Y(I+J*100)=C(I)
15810 Z(I+J*100)=D(I)
15820 R1(I+J*100)=E(I)
15830 R2(I+J*100)=F(I)
15840 Time$(I+J*100)=T$(I)
15850 IF I+J*100=Sample THEN SUBEXIT
15860 NEXT I
15870   IF J+1=Set THEN SUBEXIT
15880 NEXT J
15890 SUBEND
15910 !
```

```
15920 SUB Link(W(*),X(*),Y(*),Z(*),R1(*),R2(*),Time$(*),Count,L$)
15930 OPTION BASE 1
15940 DIM A(100),B(100),C(100),D(100),T$(100),E(100),F(100)
15950 MASS STORAGE IS ":T14"
15960 Count=0
15970 Read=1
15980 INPUT "DATA FILE TYPE? TD&RANGE1 OR TD ONLY2",Read
15990 IF (Read<>1) AND (Read<>2) THEN 15980
16000 INPUT "  FIRST FILE NAME?  ",F$
16010 L$=""
16020 Assign:ASSIGN #1 TO F$
16030   READ #1;Set,N
16040 FOR J=0 TO 3
16050 ON END #1 GOTO 16100
16060 ON Read GOTO Std,Alt
16070 Std:READ #1;A(*),B(*),C(*),D(*),E(*),F(*),T$(*)
16080 GOTO 16100
16090 Alt:READ #1;A(*),B(*),C(*),D(*),T$(*)
16100 FOR I=1 TO 100
16110 Kount=Count+I+J*100
16120 DISP Kount
16130 W(Kount)=A(I)
16140 X(Kount)=B(I)
16150 Y(Kount)=C(I)
16160 Z(Kount)=D(I)
16170 R1(Kount)=E(I)
16180 R2(Kount)=F(I)
16190 Time$(Kount)=T$(I)
16200 IF I+J*100=N THEN Status=1
16210 IF Status=1 THEN Jump
16220 IF Kount=400 THEN Jump
16230 NEXT I
16240 IF Set=J+1 THEN Jump
16250 NEXT J
16260 Jump:!
16270 PRINTER IS 16
16280 PRINT PAGE;LIN(20);"CURRENT FILE=";F$
16290 INPUT "CORRECT TD DATA? Y OR N",C$
16300 IF C$<>"Y" THEN 16460
16310 PRINT "START TIME=";Time$(Count+1);TAB(30);"STOP TIME=";Time$(Kount)
16320 INPUT " CORRECTION TO TDW(MICROSEC)?",Wc
16330 INPUT " CORRECTION TO TDX(MICROSEC)?",Xc
16340 INPUT " CORRECTION TO TDY(MICROSEC)?",Yc
16350 INPUT " CORRECTION TO TDZ(MICROSEC)?",Zc
16360 FOR I=Count+1 TO Kount
16370 W(I)=W(I)+Wc
16380 X(I)=X(I)+Xc
16390 Y(I)=Y(I)+Yc
16400 Z(I)=Z(I)+Zc
16410 NEXT I
16420 F$=F$&"t"
16430 PRINTER IS 0
16440 PRINT "FILE=";F$
16450 PRINT "TD CORRECTIONS:";TAB(20);"Wcor=";Wc;TAB(35);"Xcor=";Xc;TAB(50);"Yco
r=";Yc;TAB(65);"Zcor=";Zc
```

```
16460 Count=Kount
16470 PRINTER IS 16
16480 Status=0
16490 Comma$=CHR$(44)
16500 IF LEN(L$)<1 THEN Comma$=""
16510 L$=L$&Comma$&F$
16520 PRINT PAGE;LIN(20)
16530   PRINT "FILES LINKED=";L$
16540 PRINT "TOTAL SAMPLES=";Count
16550 IF Count=400 THEN 16610
16560 INPUT "   ANOTHER FILE?  ",A$
16570 IF A$="N" THEN End
16580 INPUT "   FILE NAME?  ",F$
16590 ! L$=L$&CHR$(44)&F$
16600 GOTO Assign
16610 End:PRINT "FILES LINKED=";L$
16620 PRINT "TOTAL SAMPLES=";Count
16630 PRINT
16640 INPUT "  STORE DATA SET?  ",S$
16650 IF S$="Y" THEN Store
16660 SUBEXIT
16670 Store:CALL Store(W(*),X(*),Y(*),Z(*),R1(*),R2(*),Time$(*),Count,F$)
16680 SUBEND
16681 !
```

```
16690 SUB Stat_mat(W(*),X(*),Y(*),Z(*),S(*),Co(*),O(*),N)
16700 OPTION BASE 1
16710 DEFAULT ON
16720 DIM A(400),B(400),C(400),D(400),V(4)
16730 ! OFF-SET
16740 O(1)=W(1)
16750 O(2)=X(1)
16760 O(3)=Y(1)
16770 O(4)=Z(1)
16780 ! SUBTRACT OFF-SET
16790 FOR I=1 TO N
16800 A(I)=W(I)-O(1)
16810 B(I)=X(I)-O(2)
16820 C(I)=Y(I)-O(3)
16830 D(I)=Z(I)-O(4)
16840 NEXT I
16850 ! SUMS
16860 S(1,1)=SUM(A)
16870 S(1,2)=SUM(B)
16880 S(1,3)=SUM(C)
16890 S(1,4)=SUM(D)
16900 ! SUMS OF SQUARES
16910 S(2,1)=DOT(A,A)
16920 S(2,2)=DOT(B,B)
16930 S(2,3)=DOT(C,C)
16940 S(2,4)=DOT(D,D)
16950 ! MEAN AND STAN DEV
16960 FOR I=1 TO 4
16970 S(3,I)=S(1,I)/N
16980 V(I)=S(2,I)/N-S(3,I)^2
16990 S(4,I)=SQR(V(I))
17000 NEXT I
17010 ! SUM OF CROSS PRODUCTS
17020 Co(1,1)=DOT(A,B)
17030 Co(1,2)=DOT(A,C)
17040 Co(1,3)=DOT(A,D)
17050 Co(1,4)=DOT(B,C)
17060 Co(1,5)=DOT(B,D)
17070 Co(1,6)=DOT(C,D)
17080 ! COVARIANCE
17090 Co(2,1)=Co(1,1)/N-S(3,1)*S(3,2)
17100 Co(2,2)=Co(1,2)/N-S(3,1)*S(3,3)
17110 Co(2,3)=Co(1,3)/N-S(3,1)*S(3,4)
17120 Co(2,4)=Co(1,4)/N-S(3,2)*S(3,3)
17130 Co(2,5)=Co(1,5)/N-S(3,2)*S(3,4)
17140 Co(2,6)=Co(1,6)/N-S(3,3)*S(3,4)
17150 ! CORRELATION COEFFICIENT
17160 Co(3,1)=Co(2,1)/(S(4,1)*S(4,2))
17170 Co(3,2)=Co(2,2)/(S(4,1)*S(4,3))
17180 Co(3,3)=Co(2,3)/(S(4,1)*S(4,4))
17190 Co(3,4)=Co(2,4)/(S(4,2)*S(4,3))
17200 Co(3,5)=Co(2,5)/(S(4,2)*S(4,4))
17210 Co(3,6)=Co(2,6)/(S(4,3)*S(4,4))
17220 SUBEND
17240 !
```

```
17250 SUB Store(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
17260 OPTION BASE 1
17270 DIM A(100),B(100),C(100),D(100),T$(100),E(100),F(100)
17280 IF N<=100 THEN Set=1
17290 IF (N>100) AND (N<=200) THEN Set=2
17300 IF (N>200) AND (N<=300) THEN Set=3
17310 IF (N>300) AND (N<=400) THEN Set=4
17320 INPUT "   FILE NAME?  ",F$
17330 PRINT "FILE=";F$,"SETS=";Set,"N=";N
17340 CREATE F$,Set,7000
17350 ASSIGN #1 TO F$
17360 PRINT #1;Set
17370 PRINT #1;N
17380 J=0
17390 Load:!
17400 FOR I=1 TO 100
17410 IF I+J*100>N THEN Zero
17420 A(I)=W(I+J*100)
17430 B(I)=X(I+J*100)
17440 C(I)=Y(I+J*100)
17450 D(I)=Z(I+J*100)
17460 E(I)=R1(I+J*100)
17470 F(I)=R2(I+J*100)
17480 T$(I)=U$(I+J*100)
17490 GOTO Next
17500 Zero:!
17510 A(I)=0
17520 B(I)=0
17530 C(I)=0
17540 D(I)=0
17550 E(I)=0
17560 F(I)=0
17570 T$(I)=""
17580 Next:NEXT I
17590 ! ON END #1 GOTO 16770
17600 ! PAUSE
17610 PRINT #1;A(*),B(*),C(*),D(*),E(*),F(*),T$(*)
17620 IF J+1=Set THEN SUBEXIT
17630 J=J+1
17640 GOTO Load
17650 SUBEND
17670 !
```

```
17680 SUB Delete_blok(W(*),X(*),Y(*),Z(*),R1(*),R2(*),T$(*),N,F$)
17690 OPTION BASE 1
17700 DIM We(400),Xe(400),Ye(400),Ze(400),Te$(400),R1e(400),R2e(400)
17710 PRINTER IS 0
17720 Sample:INPUT "FIRST SAMPLE IN BLOCK TO BE DELETED",N1
17730 INPUT "LAST SAMPLE IN BLOCK TO BE DELETED",N2
17740 J=0
17750 FOR I=1 TO N
17760 Sampl:IF (I>=N1) AND (I<=N2) THEN Next
17770 J=J+1
17780 DISP I,J
17790 We(J)=W(I)
17800 Xe(J)=X(I)
17810 Ye(J)=Y(I)
17820 Ze(J)=Z(I)
17830 R1e(J)=R1(I)
17840 R2e(J)=R2(I)
17850 Te$(J)=T$(I)
17860 Next: NEXT I
17870 N=J
17880 MAT W=We
17890 MAT X=Xe
17900 MAT Y=Ye
17910 MAT Z=Ze
17920 MAT R1=R1e
17930 MAT R2=R2e
17940 FOR I=1 TO N
17950 T$(I)=Te$(I)
17960 NEXT I
17970   PRINT "SAMPLES";N1;"THRU";N2;"DELETED,";N;"SAMPLES REMAIN"
17980 INPUT "ANOTHER BLOCK TO DELETE?",A$
17990 IF A$="Y" THEN Sample
18070 End:SUBEND
18090 !
```

```
18100 SUB Delete_td(W(*),X(*),Y(*),Z(*),R1(*),R2(*),T$(*),N,F$)
18110 OPTION BASE 1
18120 DIM We(400),Xe(400),Ye(400),Ze(400),Te$(400),R1e(400),R2e(400)
18121 Sec$="WXYZ"
18130 PRINTER IS 0
18140 Sample:N1=N
18150 INPUT "CLIP LIMITS FOR WHICH SECONDARY? W1 ,X2, Y3, Z4",Sec
18151 IF (Sec<1) OR (Sec>4) THEN Sample
18152 PRINT RPT$("_",80)
18153 PRINT "SETTING CLIP LIMITS FOR ";Sec$[Sec;1]
18154 INPUT "CLIP LIMITS?, MIN,MAX",L1,U1
18155 PRINT "UPPER=";U1;" LOWER=";L1
18180 J=0
18190 FOR I=1 TO N
18200 Sampl:ON Sec GOTO Tw,Tx,Ty,Tz
18201 Tw:IF (W(I)<L1) OR (W(I)>U1) THEN Next
18202 GOTO Keep
18203 Tx:IF (X(I)<L1) OR (X(I)>U1) THEN Next
18204 GOTO Keep
18205 Ty:IF (Y(I)<L1) OR (Y(I)>U1) THEN Next
18206 GOTO Keep
18207 Tz:IF (Z(I)<L1) OR (Z(I)>U1) THEN Next
18230 Keep:J=J+1
18240 DISP I,J
18250 We(J)=W(I)
18260 Xe(J)=X(I)
18270 Ye(J)=Y(I)
18280 Ze(J)=Z(I)
18290 R1e(J)=R1(I)
18300 R2e(J)=R2(I)
18310 Te$(J)=T$(I)
18320 Next: NEXT I
18330 N=J
18340 MAT W=We
18350 MAT X=Xe
18360 MAT Y=Ye
18370 MAT Z=Ze
18380 MAT R1=R1e
18390 MAT R2=R2e
18400 FOR I=1 TO N
18410 T$(I)=Te$(I)
18420 NEXT I
18430 PRINT N1-N;"SAMPLES DELETED,";N;"SAMPLES REMAIN"
18431 INPUT "SET ADDITIONAL CLIP LIMITS?",Q$
18432 IF UPC$(Q$[1,1])="Y" THEN Sample
18510 End:SUBEND
18511 !
```

```
18520 SUB Track(Stat(*),O_set(*),Cov(*),R(*),Sample,F$,Wt,Wf,Angle,N)
18530 DEG
18540 OPTION BASE 1
18550 DEFAULT ON
18560 FIXED 2
18570 PRINT "LORAN-C POSITION ANALYSIS: ";F$
18571 Angle=Angle-180
18572 IF Angle<0 THEN Angle=Angle+360
18580   PRINT TAB(5);"FROM WP";Wf;" TO WP";Wt;",TRACK= ";Angle;" DEGREES"
18590 PRINT "REFERENCE WAYPOINT=";Wt
18600   Slope=R(1,1)
18610   IF R(3,1)=2 THEN Slope=1/Slope
18620   A=ATN(1/Slope)
18630   IF Slope<0 THEN A=A+180
18640   IF A<0 THEN A=A+360
18650   PRINT TAB(5);"RMS TRACKLINE=";A;" DEGREES"
18660   PRINT TAB(5);"AVE CROSS TRACK DISTANCE=";(O_set(3)+Stat(3,3))*1000;" ME
ERS"
18670   PRINT TAB(5);"STD DEV CROSS TRACK DISTANCE=";Stat(4,3)*1000;" METERS"
18680 FIXED 3
18690 PRINT TAB(5);"AVE X POS=";O_set(1)+Stat(3,1);"KM";TAB(40);"Y POS=";O_set(
)+Stat(3,2);"KM"
18700 PRINT TAB(5);"STANDARD DEV X=";Stat(4,1)*1000;" METERS";TAB(40);"Y=";Stat
4,2)*1000;"METERS"
18710 SUBEND
18730 !
```

```
18740 SUB Pseudo(Glat,Glon,R(*),Plat,Plon)
18750 OPTION BASE 1
18760 RAD
18770 DIM X(3),T(3)
18780 T(1)=COS(Glat)*COS(Glon)
18790 T(2)=COS(Glat)*SIN(Glon)
18800 T(3)=SIN(Glat)
18810 MAT X=R*T
18820 H=SQR(X(1)^2+X(2)^2)
18830 Plat=ATN(X(3)/H)
18840 Plon=ATN(X(2)/X(1))
18850 SUBEND
18870 !
```

```
18880 DEF FNRange(X1,Y1,X2,Y2)
18890 RETURN SQR((X2-X1)^2+(Y2-Y1)^2)
18900 FNEND
18920 !
```

```
18930 SUB Rotate(Glat,Glon,R(*))
18940 RAD
18950 OPTION BASE 1
18960 R(1,1)=COS(Glat)*COS(Glon)
18970 R(2,1)=-SIN(Glon)
18980 R(3,1)=-SIN(Glat)*COS(Glon)
18990 R(1,2)=COS(Glat)*SIN(Glon)
19000 R(2,2)=COS(Glon)
19010 R(3,2)=-SIN(Glat)*SIN(Glon)
19020 R(1,3)=SIN(Glat)
19030 R(2,3)=0
19040 R(3,3)=COS(Glat)
19050 SUBEND
19070 !
```

```
19080 SUB Cart(Plat,Plon,X,Y)
19090 OPTION BASE 1
19100 RAD
19110 R=6378.135
19120 Cos=COS(Plat)*COS(Plon)
19130 Sin=SQR(1-Cos^2)
19140 Theta=ATN(Sin/Cos)
19150 S=R*Theta
19160 Sin=SIN(Plat)/SIN(Theta)
19170 Cos=SIN(Plon)*COS(Plat)/SIN(Theta)
19180 X=S*Cos
19190 Y=S*Sin
19200 SUBEND
19220 !
```

```
19230 DEF FNGlat(Lat,F)
19240 RAD
19250 RETURN ATN((1-F)^2*TAN(Lat))
19260 FNEND
19280 !
```

```
19290 SUB Cart_coord(X(*),Lat,Lon,Z(*))!
19300 OPTION BASE 1
19310 ! CALCULATE XY COORDINATES OF TRANSMITTERS(Z(*)) WITH RESPECT
19320 ! TO LOCAL GRID ORIGION(Lat,Lon)
19330 ! INPUT: TRANSMITTER GEODETIC POSITIONS;X(*)
19340 !        LAT AND LON OF LOCAL GRID ORIGION; Lat,Lon
19350 ! OUTPUT: TRANSMITTER XY POSITIONS; Z(*)
19360 ! SEE APL TECH NOTE SDO 5665 "EQUATIONS FOR COMPUTING.....",APR80
19370 DIM G(4,2),P(4,2),Rotate(3,3)
19380 F=.00335278
19390 Rlat=Lat*PI/180
19400 Rlat=FNGlat(Rlat,F)
19410 Rlon=Lon*PI/180
19420 CALL Rotate(Rlat,Rlon,Rotate(*))
19430 FOR I=1 TO 4
19440 G(I,1)=X(I,1)*PI/180
19450 G(I,1)=FNGlat(G(I,1),F)
19460 G(I,2)=X(I,2)*PI/180
19470 CALL Pseudo(G(I,1),G(I,2),Rotate(*),P(I,1),P(I,2))
19480 CALL Cart(P(I,1),P(I,2),Z(I,1),Z(I,2))
19490 NEXT I
19500 SUBEND
19501 !
```

```
19510 DEF FNGlat(Rlat,F)
19520 RAD
19530 RETURN ATN((1-F)^2*TAN(Rlat))
19540 FNEND
19560 !
```

```
19570 SUB Wp3(W(*),C,W,T(*),Z(*),L$)
19580 OPTION BASE 1
19590 ! SELECT THE 3 TDs TO BE USED AS THE WAYPOINT(T(*)) FOR THE
19600 ! 3-TD FEHG SOLUTION BASED ON THE CHAIN CONFIGURATION(C)
19610 ! INPUTS:
19620 !        W(*); WAYPOINT TABLE
19630 !        C; CHAIN CONFIGURATION, 1=XYZ, 2=WXY, 3=WXZ, 4=WYZ
19640 !        W; WAYPOINT NUMBER
19650 ! OUTPUTS:
19660 !        T(*); WAYPOINT TDs
19670 !        Z(*); WAYPOINT XY POSITION
19680 !        L$; LABEL
19690     Z(1,1)=W(W,5)
19700     Z(2,1)=W(W,6)
19710 ON C GOTO Xyz,Wxy,Wxz,Wyz
19720 Xyz:!
19730     T(1,1)=W(W,2)
19740     T(2,1)=W(W,3)
19750     T(3,1)=W(W,4)
19760     L$="XYZ"
19770     SUBEXIT
19780 Wxy:     !
19790     T(1,1)=W(W,1)
19800     T(2,1)=W(W,2)
19810     T(3,1)=W(W,3)
19820     L$="WXY"
19830     SUBEXIT
19840 Wxz:!
19850     T(1,1)=W(W,1)
19860     T(2,1)=W(W,2)
19870     T(3,1)=W(W,4)
19880     L$="WXZ"
19890     SUBEXIT
19900 Wyz:     !
19910     T(1,1)=W(W,1)
19920     T(2,1)=W(W,3)
19930     T(3,1)=W(W,4)
19940     L$="WYZ"
19950 SUBEND
19970 !
```

```
19980 SUB Td3(W(*),X(*),Y(*),Z(*),I,C,T(*))
19990 OPTION BASE 1
20000 ! SELECTS THE PROPER TD SAMPLES FOR CALCULATING XY POSITION
20010 ! BASED ON CHAIN CONFIGURATION AND 3-TD FEHG SOLUTION
20020 ! INPUTS:
20030 !       W(*),X(*),Y(*),Z(*); TD DATA ARRAYS
20040 !       I; SAMPLE NUMBER
20050 !       C; CHAIN CONFIGURATION 1=XYZ, 2=WXY, 3=WXZ, 4=WYZ
20060 ! OUTPUT:
20070 !       T(*); TD SAMPLE
20080 ON C GOTO Xyz,Wxy,Wxz,Wyz
20090 Xyz:!
20100     T(1,1)=X(I)
20110     T(2,1)=Y(I)
20120     T(3,1)=Z(I)
20130     SUBEXIT
20140 Wxy:    !
20150    T(1,1)=W(I)
20160    T(2,1)=X(I)
20170    T(3,1)=Y(I)
20180    SUBEXIT
20190 Wyz:   !
20200    T(1,1)=W(I)
20210    T(2,1)=Y(I)
20220    T(3,1)=Z(I)
20230    SUBEXIT
20240 Wxz:   !
20250    T(1,1)=W(I)
20260    T(2,1)=X(I)
20270    T(3,1)=Z(I)
20280 SUBEND
20300 !
```

```
20310 SUB Wp2(W(*),C,W,T(*),P,Z(*),G12(*),G23(*),G13(*),G(*),L$,Zx(*),Zxt(*))
20320 OPTION BASE 1
20330 ! SELECT WAYPOINT TDs,POSITION,TRANSMITTER POSITIONS AND G-MATRIX FOR 2-TD
 FEHG SOLUTION
20340 ! FEHG SOLUTION BASED ON CHAIN CONFIGURATION AND DESIRED TD PAIR
20350 ! INPUTS:
20360 !        W(*); WAYPOINT TABLE
20370 !        C; CHAIN CONFIGURATION 1=XYZ, 2=WXY, 3=WXZ, 4=WYZ
20380 !        P; TD PAIR
20390 !        G12(*),G23(*),G13(*); 2-TD G-MATRICES
20400 !        Zx(*); TRANSMITTER POSITIONS
20410 ! OUTPUTS:
20420 !        T(*); WAYPOINT TDs
20430 !        G(*); G-MATRIX
20440 !        Z(*); WAYPOINT POSITION
20450 !        Zxt(*); TRANSMITTER POSITIONS FOR 2-TD SOLUTION
20460 !        L$; LABEL
20470 Z(1,1)=W(W,5)
20480 Z(2,1)=W(W,6)
20490 Zxt(3,1)=Zx(4,1)
20500 Zxt(3,2)=Zx(4,2)
20510 ON C GOTO Xyz,Wxy,Wxz,Wyz
20520 Xyz:!
20530 INPUT "INPUT TD PAIR; XY1, XZ2, YZ3",P
20540 ON P GOTO Xy,Xz,Yz
20550 Wxy:!
20560 INPUT "INPUT TD PAIR; WX1, WY2, XY3",P
20570 ON P GOTO Wx,Wy,Xy
20580 Wxz:!
20590 INPUT "INPUT TD PAIR; WX1, WZ2, XZ3",P
20600 ON P GOTO Wx,Wz,Xz
20610 Wyz:!
20620 INPUT "INPUT TD PAIR; WY1, WZ2, YZ3",P
20630 ON P GOTO Wy,Wz,Yz
20640 Wx:!
20650   T(1,1)=W(W,1)
20660   T(2,1)=W(W,2)
20670   L$="WX"
20680   GOTO Xmit_sel
20690 Wy:  !
20700   T(1,1)=W(W,1)
20710   T(2,1)=W(W,3)
20720   L$="WY"
20730   GOTO Xmit_sel
20740 Wz:  !
20750   T(1,1)=W(W,1)
20760   T(2,1)=W(W,4)
20770   L$="WZ"
20780   GOTO Xmit_sel
20790 Xy:  !
20800   T(1,1)=W(W,2)
20810   T(2,1)=W(W,3)
20820   L$="XY"
20830   GOTO Xmit_sel
20840 Xz:  !
```

```
20850   T(1,1)=W(W,2)
20860   T(2,1)=W(W,4)
20870   L$="XZ"
20880   GOTO Xmit_sel
20890 Yz:  !
20900   T(1,1)=W(W,3)
20910   T(2,1)=W(W,4)
20920   L$="YZ"
20930   GOTO Xmit_sel
20940 Xmit_sel:  !
20950 ON P GOTO X12,X13,X23
20960 X12:!
20970    MAT G=G12
20980    Zxt(1,1)=Zx(1,1)
20990    Zxt(1,2)=Zx(1,2)
21000    Zxt(2,1)=Zx(2,1)
21010    Zxt(2,2)=Zx(2,2)
21020    SUBEXIT
21030 X13:   !
21040    MAT G=G13
21050    Zxt(1,1)=Zx(1,1)
21060    Zxt(1,2)=Zx(1,2)
21070    Zxt(2,1)=Zx(3,1)
21080    Zxt(2,2)=Zx(3,2)
21090    SUBEXIT
21100 X23:   !
21110    MAT G=G23
21120    Zxt(1,1)=Zx(2,1)
21130    Zxt(1,2)=Zx(2,2)
21140    Zxt(2,1)=Zx(3,1)
21150    Zxt(2,2)=Zx(3,2)
21160   SUBEND
21180 !
```

```
21190 SUB Td2(W(*),X(*),Y(*),Z(*),I,C,P,T(*))
21200 ! SELECT THE PROPER TD SAMPLES FOR CALCULATING XY POSITION
21210 ! BASED ON CHAIN CONFIGURATION, TD PAIR, AND 2-TD FEHG SOLUTION
21220 ! INPUTS:
21230 !       W(*),X(*),Y(*),Z(*); TD DATA ARRAYS
21240 !       I; SAMPLE NUMBER
21250 !       C; CHAIN CONFIGURATION
21260 !       P; TD PAIR
21270 ! OUTPUT:
21280 !       T(*); TD SAMPLE
21290 OPTION BASE 1
21300 ON C GOTO Xyz,Wxy,Wxz,Wyz
21310 Xyz:!
21320 ON P GOTO Xy,Xz,Yz
21330 Wxy:!
21340 ON P GOTO Wx,Wy,Xy
21350 Wxz:!
21360 ON P GOTO Wx,Wz,Xz
21370 Wyz:!
21380 ON P GOTO Wy,Wz,Yz
21390 Wx:!
21400 T(1,1)=W(I)
21410 T(2,1)=X(I)
21420 SUBEXIT
21430 Wy:!
21440 T(1,1)=W(I)
21450 T(2,1)=Y(I)
21460 SUBEXIT
21470 Wz:!
21480 T(1,1)=W(I)
21490 T(2,1)=Z(I)
21500 SUBEXIT
21510 Xy:!
21520 T(1,1)=X(I)
21530 T(2,1)=Y(I)
21540 SUBEXIT
21550 Xz:!
21560 T(1,1)=X(I)
21570 T(2,1)=Z(I)
21580 SUBEXIT
21590 Yz:!
21600 T(1,1)=Y(I)
21610 T(2,1)=Z(I)
21620 SUBEND
21640 !
```

```
21650 SUB Wpf2(W(*),C,P,W,T(*))
21660 ! SELECTS PROPER TDs FROM WAYPOINT FILE FOR CALCULATION OF
21670 ! THE "FROM" WAYPOINT POSITION USING THE 2-TD FEHG SOLUTION
21680 ! INPUTS:
21690 !        W(*); WAYPOINT FILE
21700 !        C; CHAIN CONFIGURATION
21710 !        P; TD PAIR
21720 !        W; WAYPOINT FROM
21730 ! OUTPUT:
21740 !        T(*); 2-TD WAYPOINT FOR WAYPOINT FROM
21750 ON C GOTO Xyz,Wxy,Wxz,Wyz
21760 Xyz: ON P GOTO Xy,Xz,Yz
21770 Wxy: ON P GOTO Wx,Wy,Xy
21780 Wyz: ON P GOTO Wy,Wz,Yz
21790 Wx:!
21800 T(1,1)=W(W,1)
21810 T(2,1)=W(W,2)
21820 SUBEXIT
21830 Wy:!
21840 T(1,1)=W(W,1)
21850 T(2,1)=W(W,3)
21860 SUBEXIT
21870 Wz:!
21880 T(1,1)=W(W,1)
21890 T(2,1)=W(W,4)
21900 SUBEXIT
21910 Xy:!
21920 T(1,1)=W(W,2)
21930 T(2,1)=W(W,3)
21940 SUBEXIT
21950 Xz:!
21960 T(1,1)=W(W,2)
21970 T(2,1)=W(W,4)
21980 SUBEXIT
21990 Yz:!
22000 T(1,1)=W(W,3)
22010 T(2,1)=W(W,4)
22020 SUBEND
22040 !
```

```
22050 SUB Rb(X1,Y1,X2,Y2,B,R)
22060 DEG
22070 DEFAULT ON
22080 Dx=X2-X1
22090 Dy=Y2-Y1
22100 R=SQR(Dx^2+Dy^2)
22110 B=ATN(Dx/Dy)
22120 IF Dy<0 THEN B=B+180
22130 IF B<0 THEN B=B+360
22140 SUBEND
22141 !
```

```
22720  SUB Project(W,Wpt(*),W(*),X(*),Y(*),Z(*),Zx(*),Zy(*),Conf,Zxmit(*),N,V)
22730  OPTION BASE 1
22740  DIM T1(400),T2(400),T3(400),Bear(4),Range(4),T(3)
22750  !  REFLECTS TDs MEASURED NEAR A WAYPOINT TO THE WAYPOINT
22760  !  TDreflect=TDmeas-(h(Zmeas)-h(Zwaypoint))
22770  !    h(Z)=(Range_secondary-Range_master)/Velocity_propogation
22780  !              CALCULATE h(Zwaypoint)
22790  FOR I=1 TO 4
22800  CALL Rb(Wpt(W,5),Wpt(W,6),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I))
22810  NEXT I
22820  FOR I=1 TO 3
22830  T(I)=(Range(I)-Range(4))/V
22840  NEXT I
22850  !              CALCULATE h(Zmeas)-h(Zwaypoint)
22860  FOR I=1 TO N
22870  FOR J=1 TO 4
22880  CALL Rb(Zx(I),Zy(I),Zxmit(J,1),Zxmit(J,2),Bear(J),Range(J))
22890  NEXT J
22900  T1(I)=(Range(1)-Range(4))/V-T(1)
22910  T2(I)=(Range(2)-Range(4))/V-T(2)
22920  T3(I)=(Range(3)-Range(4))/V-T(3)
22930  NEXT I
22940  !              CALCULATE TDmeas-TDproj
22950  !              CALCULATE TDmeas-[TDwaypoint+(h(Zmeas)-h(Zwaypoint))]
22960  ON Conf GOTO Xyz,Wxy,Wxz,Wyz
22970 Xyz: !
22980   MAT W=(0)
22990   MAT X=X-T1
23000   MAT X=X-(Wpt(W,2))
23010   MAT Y=Y-T2
23020   MAT Y=Y-(Wpt(W,3))
23030   MAT Z=Z-T3
23040   MAT Z=Z-(Wpt(W,4))
23050   SUBEXIT
23060 Wxy: !
23070   MAT W=W-T1
23080   MAT W=W-(Wpt(W,1))
23090   MAT X=X-T2
23100   MAT X=X-(Wpt(W,2))
23110   MAT Y=Y-T3
23120   MAT Y=Y-(Wpt(W,3))
23130   MAT Z=(0)
23140   SUBEXIT
23150 Wxz: !
23160   MAT W=W-T1
23170   MAT W=W-(Wpt(W,1))
23180   MAT X=X-T2
23190   MAT X=X-(Wpt(W,2))
23200   MAT Y=(0)
23210   MAT Z=Z-T3
23220   MAT Z=Z-(Wpt(W,4))
23230   SUBEXIT
23240 Wyz: !
23250   MAT W=W-T1
23260   MAT W=W-(Wpt(W,1))
```

```
23270  MAT X=(0)
23280  MAT Y=Y-T2
23290  MAT Y=Y-(Wpt(W,3))
23300  MAT Z=Z-T3
23310  MAT Z=Z-(Wpt(W,4))
23320  SUBEND
23321  !
```

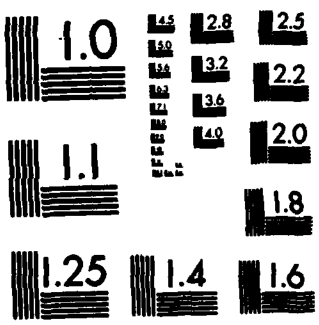
```
23330 SUB Way(S1(*),S2(*),R1(*),R2(*),N,M,Wp(*))
23340 DEG
23350 OPTION BASE 1
23360 DIM M(2,2),G(2,2),E(2,1),W(2,1)
23370 DEFAULT ON
23380 FOR I=1 TO 6
23390 ! CALCULATE WAYPOINT
23400    M(1,2)=M(2,2)=-1
23410 M(1,1)=R1(1,I)
23420 M(2,1)=R2(1,I)
23430 E(1,1)=M(1,1)*S1(1,I)-S1(2,I)
23440 E(2,1)=M(2,1)*S2(1,I)-S2(2,I)
23450 MAT G=INV(M)
23460 MAT W=G*E
23470 ! STORE WAYPOINT
23480 Wp(1,I)=W(1,1)
23490 Wp(2,I)=W(2,1)
23500 IF (S1(1,I)=0) OR (S1(2,I)=0) OR (S2(1,I)=0) OR (S2(2,I)=0) THEN Wp(1,I)=0
23510 IF (S1(1,I)=0) OR (S1(2,I)=0) OR (S2(1,I)=0) OR (S2(2,I)=0) THEN Wp(2,I)=0
23520 ! CALCULATE CROSSING ANGLES
23530 An=ATN(M(1,1))
23540 Bn=ATN(M(2,1))
23550 Cn=ABS(Bn-An)
23560 ! PRINT An,Bn,Cn
23570 ! IF Cn>90 THEN Cn=180-Cn
23580 ! CALCULATE DISTANCE BETWEEN WP AND MEAN OF INDEPENDENT VAR
23590 D1=S1(R1(3,I),I)-W(R1(3,I),1)
23600 D2=S2(R2(3,I),I)-W(R2(3,I),1)
23610 ! STANDARD DEVIATION OF IND VAR
23620 S1=S1(R1(3,I)+2,I)
23630 S2=S2(R2(3,I)+2,I)
23640 ! VAR OF REGRESSION LINE AT WP
23650 V1=R1(2,I)^2*(1/N+D1^2/(N-2)/S1^2)
23660 V2=R2(2,I)^2*(1/M+D2^2/(M-2)/S2^2)
23670 ! RMS ERROR OF WAYPOINT
23680 V=V1+V2
23690 S=SQR(V)
23700 S=S/SIN(Cn)
23710 Wp(3,I)=S
23720 Wp(4,I)=Cn
23730 NEXT I
23740 SUBEND
23760 !
```

```
23770 SUB Wprint(Wp(*))
23780 OPTION BASE 1
23790 FOR I=1 TO 4
23800 FOR J=1 TO 6
23810 IF Wp(I,J)>10^6 THEN Wp(I,J)=0
23820 NEXT J
23830 NEXT I
23840  PRINT RPT$("_",80)
23850  PRINT LIN(1);TAB(30);"WAYPOINT SOLUTION"
23860  PRINT RPT$("_",80)
23870  FIXED 3
23880  PRINT LIN(1);TAB(1);"TD PAIR";TAB(20);"WX";TAB(31);"WY";TAB(42);"WZ";TAB(
53);"XY";TAB(64);"XZ";TAB(75);"YZ"
23890  PRINT LIN(1);TAB(1);"TDW";TAB(15);Wp(1,1);TAB(26);Wp(1,2);TAB(37);Wp(1,3)
23900  PRINT LIN(1);TAB(1);"TDX";TAB(15);Wp(2,1);TAB(48);Wp(1,4);TAB(59);Wp(1,5)
23910  PRINT LIN(1);TAB(1);"TDY";TAB(26);Wp(2,2);TAB(48);Wp(2,4);TAB(70);Wp(1,6)
23920  PRINT LIN(1);TAB(1);"TDZ";TAB(37);Wp(2,3);TAB(59);Wp(2,5);TAB(70);Wp(2,6)
23930  PRINT LIN(1);TAB(1);"RMS ERROR";TAB(18);Wp(3,1);TAB(29);Wp(3,2);TAB(40);W
p(3,3);TAB(51);Wp(3,4);TAB(62);Wp(3,5);TAB(73);Wp(3,6)
23940  PRINT LIN(1);TAB(1);"CROSSING ANGLE";TAB(18);Wp(4,1);TAB(29);Wp(4,2);TAB(
40);Wp(4,3);TAB(51);Wp(4,4);TAB(62);Wp(4,5);TAB(73);Wp(4,6)
23950  PRINT RPT$("_",80)
23960  PRINT LIN(1)
23970  SUBEND
23971  !
```

```
23980  SUB Data(W(*),X(*),Y(*),Z(*),N,Ang,Ds,De,Sigma,Xmit(*),Zp(*),V,Conf)
23990  DEG
24000  OPTION BASE 1
24010  DEFAULT ON
24020  DIM R(4),T1(400),T2(400),T3(400),T(3),Zq(2)
24030  !       CALCULATE SAMPLE INTERVAL
24040  Di=(De-Ds)/N
24050  !       CALCULATE TD SAMPLES
24060  D=Ds
24070  !               CALCULATE h(Zp(*))
24080  FOR J=1 TO 4
24090  CALL Range(Zp(1),Zp(2),Xmit(J,1),Xmit(J,2),R(J))
24100  NEXT J
24110  FOR J=1 TO 3
24120  T(J)=(R(4)-R(J))/V
24130  NEXT J
24140  FOR I=1 TO N
24150  DISP I;N
24160  !               CALCULATE Zq(*)
24170  DEG
24180  Zq(1)=D*COS(Ang)/(1.852*60)+Zp(1)               ! LAT
24190  Zq(2)=D*SIN(Ang)/(1.852*60*COS(Zp(1)))+Zp(2)  ! LON
24200  !               INCREMENT D
24210  D=D+Di
24220  !               CALCULATE NOISE
24230  RAD
24240  U1=RND
24250  U2=RND
24260  U3=RND
24270  N1=(-2*LOG(U1))^.5*COS(2*PI*U2)*Sigma
24280  N2=(-2*LOG(U1))^.5*SIN(2*PI*U2)*Sigma
24290  N3=(-2*LOG(U2))^.5*SIN(2*PI*U3)*Sigma
24300  !               CALCULATE h(Zq)-h(Zp)+N
24310  FOR J=1 TO 4
24320  CALL Range(Zq(1),Zq(2),Xmit(J,1),Xmit(J,2),R(J))
24330  NEXT J
24340  T1(I)=(R(4)-R(1))/V-T(1)+N1
24350  T2(I)=(R(4)-R(2))/V-T(2)+N2
24360  T3(I)=(R(4)-R(3))/V-T(3)+N3
24370  NEXT I
24380  !              STORE DATA IN PROPER ARRAY
24390  ON Conf GOTO Xyz,Wxy,Wxz,Wyz
24400 Xyz: !
24410  MAT W=(0)
24420  MAT X=T1
24430  MAT Y=T2
24440  MAT Z=T3
24450  SUBEXIT
24460 Wxy: !
24470 MAT W=T1
24480 MAT X=T2
24490 MAT Y=T3
24500 MAT Z=(0)
24510 SUBEXIT
24520 Wxz:!
```

AD-A124 343 HHE/LORAN-C SURVEYING(U) COAST GUARD WASHINGTON DC
OFFICE OF RESEARCH AND DEVELOPMENT A J SEDLOCK NOV 82 3/4
USCG-D-54-82

UNCLASSIFIED F/G 17/7 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
24530  MAT W=T1
24540  MAT X=T2
24550  MAT Y=(0)
24560  MAT Z=T3
24570  SUBEXIT
24580 Wyz: !
24590  MAT W=T1
24600  MAT X=(0)
24610  MAT Y=T2
24620  MAT Z=T3
24630  SUBEND
24631  !
```

```
24640 SUB Read_alt(W(*),X(*),Y(*),Z(*),Time$(*),Sample,F$)
24650 OPTION BASE 1
24660 DIM A(100),B(100),C(100),D(100),T$(100)
24670 MASS STORAGE IS ":T14"
24680 Sample=0
24690 MAT W=(0)
24700 MAT X=(0)
24710 MAT Y=(0)
24720 MAT Z=(0)
24730 FOR I=1 TO 400
24740 Time$(I)=""
24750 NEXT I
24760 INPUT "  FILE NAME? ",F$
24770 ASSIGN #1 TO F$
24780 ! READ #1;Set
24790   READ #1;Set,Sample
24800 FOR J=0 TO 3
24810 ON END #1 GOTO 24830
24820 READ #1;A(*),B(*),C(*),D(*),T$(*)
24830 FOR I=1 TO 100
24840 W(I+J*100)=A(I)
24850 X(I+J*100)=B(I)
24860 Y(I+J*100)=C(I)
24870 Z(I+J*100)=D(I)
24880 Time$(I+J*100)=T$(I)
24890  ! IF ((A(I)=B(I))=C(I))=D(I) THEN Sample=I-1+J*100
24900 ! IF Sample<>0 THEN SUBEXIT
24910 IF I+J*100=Sample THEN SUBEXIT
24920 NEXT I
24930   IF J+1=Set THEN SUBEXIT
24940 NEXT J
24950 SUBEND
24960 !
```

```
24970 SUB Separate(W(*),X(*),Y(*),Z(*),R1(*),R2(*),T$(*),Wt(*),Xt(*),Yt(*),Zt(*)
,R1t(*),R2t(*),Tt$(*),N)
24980 OPTION BASE 1
25000 N=0
25010 MAT Wt=(0)
25020 MAT Xt=(0)
25030 MAT Yt=(0)
25040 MAT Zt=(0)
25041 MAT R1t=(0)
25042 MAT R2t=(0)
25043 FOR I=1 TO 400
25044 Tt$(I)=""
25045 NEXT I
25050 PRINTER IS 16
25060 PRINT LIN(20);"FOR SUBFILE ENTER:"
25070 INPUT "START SAMPLE NUMBER",Start
25080 INPUT "STOP SAMPLE NUMRER",Stop
25090 FOR I=Start TO Stop
25100 N=N+1
25110 Wt(N)=W(I)
25120 Xt(N)=X(I)
25130 Yt(N)=Y(I)
25140 Zt(N)=Z(I)
25150 R1t(N)=R1(I)
25160 R2t(N)=R2(I)
25170 Tt$(N)=T$(I)
25180 NEXT I
25210 SUBEND
25211 !
```

```
25220 DEF FNReady!
25230 A=0
25240 COM Dcom$[500],Sel_code
25250 IF POS(Dcom$,CHR$(13))>0 THEN A=1
25260 RETURN A
25261 !
```

APPENDIX B

COMPAR, Data Analysis Software for Electronic Positioning Augmentation

DOCUMENTATION FOR PROGRAM "COMPAR"

TABLE OF CONTENTS

PROGRAM APPLICATIONS

The program "COMPAR" is a collection of special function programs used to analyze data collected during a Loran-C Harbor Survey with microwave positioning augmentation. The major functions of the program are to:

a. Calculate waypoint Loran-C time differences

b. Evaluate the performance of Loran-C positioning against the microwave reference system (e.g. Mini-ranger)

c. Calculate Loran-C time differences of trackpoints

The following is a set of general procedures to use in applying COMPAR to analyze data collected during a harbor survey with microwave positioning augmentation:

Preliminaries. Waypoint table, Loran-C chain data, and Mini-ranger reference station data files are necessary for complete data analysis. The waypoint table file contains time-differences, local xy coordinates, latitude and longitude for twenty-five waypoints (see K23). The Loran-C chain data file contains latitude and longitude and power level for three secondaries and the master and the emission delays for the three secondaries (see K26). The Mini-ranger reference station files contain the name and xy coordinates of two reference stations and the local origin used to calculate position using Mini-ranger data (see K28). The waypoint table, Loran-C chain data, and Mini-ranger reference station data files are prepared during the survey planning stages. It is convenient to have the waypoint table file(s), chain data file and reference station file(s) on the same data tape cartridge.

The Loran-C TD data may be corrected for known offsets during the time period that data was collected. Offsets can be determined from a local monitor TD record and data from the system area monitor. Known calibration errors in the Mini-ranger range data may also be corrected (see K0).

Waypoint TD Calculations: The primary method to calculate waypoint TDs is to "reflect" TDs collected near a waypoint to the waypoint. The "reflect" function (see K27) calculates the difference in TDs between the measurement position and the waypoint position. This difference is applied as a correction to the measured TDs for each sample. The mean and standard deviation of the "reflected" TDs are then calculated.

The standard deviation of the reflected TDs should be slightly larger than the standard deviations of data collected at dockside. Outliers in the Mini-ranger data is the usual cause of high standard deviations in the reflected TD data. It is a good practice to plot the xy positions (see K18) calculated from the Mini-ranger data prior to reflecting the data to the waypoint. Outliers are easily identified from the plotted data. Another suggestion is to perform the automatic range outlier edit function (see K16, option 4) prior to converting the range data to xy positions.

Data is collected at several points near a waypoint (typically moored to or stationed near a buoy in the vicinity of the waypoint). An optional data collection technique is to maneuver the survey vessel in a cloverleaf pattern centered at the waypoint. Waypoint TD estimates are calculated from each data file. The results are compared to each as a check for errors. Agreement to within 20 to 30 nanoseconds of the mean can be expected. Differences are the results of estimation errors due to noise, uncorrected TD offsets, the difference in distance between the measurement positions and the waypoint, uncorrected range errors, etc. It is left to the judgement of the data analyst how to combine the waypoint TD estimates for each data file to form the final TD estimate. The most straightforward approach is to average the estimates. Figure 1 is a flow chart for calculating waypoints TD.

For each data file collected near the waypoint of interest

    Read data file (K0)

        Correct range data
        Correct TD data

    Edit data for range errors (K16, option 4)

    Convert range data to xy position (K1)

    Plot xy data and check for outliers (K18)

    If outliers are detected:

        Delete bad samples (K16, options 1 or 2

        AND

        Recalculate xy position (K1)

    Reflect TDs to waypoint of interest (K27)

Calculate final waypoint estimate (i.e. average results from each data file)

FLOWCHART OF WAYPOINT CALCULATIONS

FIGURE 1

Comparison of Performance. TD and Mini-ranger data are collected along channel edges and centerlines during a harbor survey with microwave positioning augmentation. These data files can be used to compare the position fixes obtained from Mini-ranger and the positions calculated from the TDs using calculated waypoints. This comparison provides a verification of the waypoint calculations, a measure of performance which may be expected using a Loran-C navigation device (e.g. PILOT), and enables the data analyst to decide if a trackpoint is necessary between waypoints.

The coordinates for waypoints and Mini-ranger reference stations are based on Army Corps of Engineers (COE) dredging data. The COE coordinates are in state plane coordinates. These coordinates are translated to local coordinates by subtracting the state plane coordinates of the local origin and changing units from feet to kilometers. Waypoints are calculated as the intersection of channel centerlines.

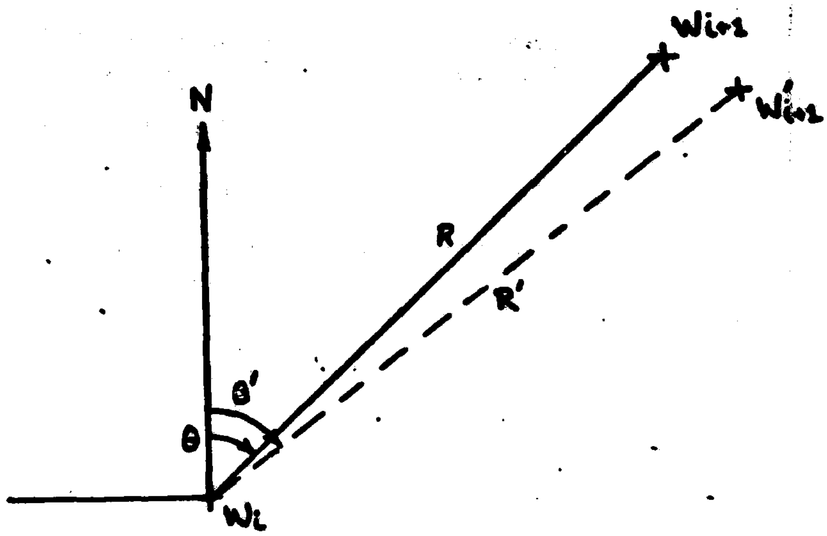
The position fixes for Mini-ranger data are computed using trilateration (see K1). Along and crosstrack positions are based on the range and bearing calculated between tabulated waypoint coordinates.

The Flat Earth Hyperbolic Grid (FEHG) algorithm is used to compute xy coordinates from TD data (see K17). Inputs to the algorithm are calculated waypoint TDs, waypoint xy coordinates and transmitter xy coordinates. The transmitter xy coordinates are referenced to the local origin. The FEHG algorithm is also used to calculate the coordinates of the next waypoint for along/cross track calculations.

The function "Compare Mini-ranger and Loran-C Position Data" (see K3) calculates the difference in xy position and along/cross track position for each data sample and the mean and rms values of the differences for the data set. If the Mini-ranger positions are assumed to be "truth," the result is an evaluation of the accuracy of the Loran-C fixes for the data set. Plots are provided for the xy "errors" and along/cross track "errors." Plots of the xy data and along/cross track data are also available (see K18 and K19).

One must be careful to compare "oranges to oranges" when evaluating the "error" statistics and plots generated by the function K3. The domain in which a Loran-C harbor navigation device (e.g. PILOT) operates is distances relative to a trackline defined by two waypoints (i.e. along/cross track). The xy "error" data is interesting, but its only significance is that the x and y errors should approach zero near the "TO" waypoint (along track distance = 0). Large xy errors near the "TO" waypoint is indicative of an error in the waypoint, the Mini-ranger data, or the Loran-C data.

The along/cross Drack "errors" are the more significant. If both the waypoint "TO" and "FROM" waypoint calculations are correct, the cross track distance "errors" at each end of the trackline will be near zero. The along track distance errors should converge to zero at the "TO" waypoint. Note that for the Mini-ranger data, along and cross track distance are calculated using the distance and bearing calculated between the waypoint coordinates determined from COE dredging data (see Figure 2). For the Loran-C data, the "TO" waypoint xy coordinates are the same as those used in the Mini-ranger calculations. The "FROM" waypoint coordinates (Wi+1 in Figure 2) are calculated using the FEHG algorithm and the waypoint TDs. The range and bearing between Wi and Wi+1 (R and -) are not, in general, equal to the range and bearing between Wi and Wi+1 (R and -).



Wi - Waypoint "TO" xy position; calculated from COE dredging data; stored in waypoint table

Wi+1 - Waypoint "FROM" xy position; calculated from COE dredging data; stored in waypoint table

W'i+1 - Waypoint "FROM" xy position calculated using FEHG algorithm relative to Wi

O O bearing from Wi to Wi+1

O' - bearing from Wi to Wi+1

R - range from Wi to Wi+1

R' - range from Wi to W'i+1

FIGURE 2

Read data file (K0)

Correct range data
Correct TD data

Edit data for range errors (K16, option 4)

Convert range data to xy position (K1)

Convert TD data to xy position (K17)
(Use same "TO" and "FROM" waypoints as above)

Compare the two sets of xy data (K3)

Plot xy data (K18) - optional

Plot along/cross track data (K19) - optional

If cross track errors are significant

Calculate grid warp (K21)
Calculate track point

SUMMARY OF PROCEDURE TO COMPARE POSITION DATA
CALCULATED FROM MINI-RANGER AND LORAN-C

FIGURE 3

Trackpoint Calculations. Cross track error is the best measure of the expected performance of Loran-C navigation between waypoints. The plot of the cross track errors may exhibit a "bow" between the waypoints. That is, the error will be near zero at both endpoints, and there will be a well defined offset somewhere near the halfway point. The severity of this bow and the channel dimensions are the factors which must be evaluated to determine if a track point is necessary. A trackpoint is basically a waypoint located on the centerline between two waypoints. The data analyst determines the location of a trackpoint(s) by inspecting the cross track error plot. The along track distance where a trackpoint(s) is needed is recorded. The function, Calculate Grid Warp (see K21), is used to calculate the difference between the measured TDs and the TDs projected from the waypoint based on the difference in position between the waypoint and measurement point. The differences are plotted against along track distance. Therefore, the correction to be applied at the trackpoint for each TD can be picked off the error plot at the along track distance the trackpoint is to be located. The ideal TD at the trackpoint is calculated using the TD Move (see K24) function. The trackpoint TD is equal to the ideal TD plus the correction determined from the TD error plot.

Miscellaneous. The mean and standard deviation of TD data collected are calculated using the function, Stats and Regression of TD Data (see K2). This function also calculates covariance, correlation coefficients, and linear regression slopes for each TD pair. The TD data is plotted using the function, Plot TD Data with Regression Lines (see K4). Residuals for the linear regression of TD pairs is plotted using the function, Plot Residuals of TD data (see K5).

The function, Predict TD (see K20), is used to predict the TDs for a waypoint or any position of interest. The function also outputs ranges and bearings to the transmitters and geometric dilution of precision (GDOP) for the three TD fix and each of the possible two TD fixes.

The function, Link Data Files (see K25), is used to load more than one data file into computer memory. The function will limit the total number of samples at 400. The combined data may then be restored on a new file.

The function, Daisy Chain Waypoints (see K25), calculates the position of one waypoint with respect to another based on the difference in TDs between waypoints. This function is more frequently used during visual surveys. However, it is very useful to compare the range and bearing computed with the range and bearing from COE coordinates as a check of waypoint calculations. Bearing angles are generally within .5 degrees, and ranges within 100 meters.

The functions, File or Read WP Data (see K23), Store Chain Data (see K26), and File or Read Reference Station Data (see K28), are used to file reference data for further use and to store waypoint data.

Function: Read Data File
Special function key: K0
Subprograms:

Read (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)

The program function reads time-difference (TD), range and time-of-day data stored on magnetic tape. The range and TD data may be corrected for known errors. If the range data is corrected, a lower case "r" is annotated to the file name, F$. If TDs are corrected, a lower case "t" is annotated to F$.

Input parameters:

- F$ - File name
- Range $ - Indicates if Range data is to be corrected, "Y" = yes, "N" = no.
- R1c - correction to R1(*)
- R2c - correction to R2(*)
- Td_cor$ - Indicates if TD data is to be corrected, "Y" = yes, "N" = no.
- Wcor - correction to W(*)
- Xcor - correction to X(*)
- Ycor - correction to Y(*)
- Zcor - correction to Z(*)

Output parameters:

- W(*) - TDW data
- X(*) - TDX data
- Y(*) - TDY data
- Z(*) - TDZ data
- R1(*) - Range 1 data
- R2(*) - Range 2 data
- U$(*) - Time-of-day data
- N - number of samples
- F$ - annotated file name.
  - F$&"r" = range data corrected
  - F$&"t" = TD data corrected
  - F$&"rt" = range and TD data corrected.

User Instructions:

Prerequisite functions: None

1. Insert data tape into left hand tape transport.

2. Press K0. The menu is cleared and "READ DATA FILE" appears on the CRT in inverse video.

3. When "File Name?" appears in display area:

   a. Enter: File name
   b. Press: CONT

The file is read and the file name and number of samples are printed.

4. When "CORRECT RANGE DATA? Y OR N" appears in the display area:

a. If you want to correct range data:
   (1) Enter: Y
   (2) Press: CONT
   (3) Go to Step 5
   or
b. If you do not want to correct range data:
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 7.

5. When "CORRECTION TO R1 (METERS)?" appears in display area:

a. Enter: correction to Range 1 in meters
b. Press: CONT

6. When "CORRECTION TO R2(METERS)? appears in display area:

a. Enter: correction to Range 2 in meters
b. Press: CONT

The corrections entered for Range 1 and Range 2 are printed and titled.

7. When "CORRECTION TO TDs? Y/N" appears in display area:

a. If you want to correct TD data:

   (1) Enter: Y
   (2) Press: CONT
   (3) Go to Step 8
b. If you do not want to correct TD data:

   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 12

8. The start and stop times for the data collected will be printed. When "CORRECTION TO TDW(MICROSEC)?" appears in display area:

a. Enter: correction to TDW data in microseconds
b. Press: CONT

9. When "CORRECTION TO TDX(MICROSEC)?" appears in display area:

a. Enter: correction to TDX data in microseconds
b. Press: CONT

10. When "CORRECTION TO TDY(MICROSEC)?" appears in display area:

a. Enter: correction to TDY data in microseconds
b. Press: CONT

11. When "CORRECTION TO TDZ(MICROSEC)?" appears in display area:

a. Enter: correction to TDZ data in microseconds
b. Press: CONT

The corrections entered for each TD will be printed and titled.

12. The program function is complete. The function menu is printed on the CRT.

Function: Convert Range Data to XY Positions
Special Function Key: K1
Subprograms:

Triangle (R, Rb/1000, Rc/1000, A, B, C)
Position (Zb(*), Alpha, Rc/1000, Sign, B, Zx(I), Zy(I))
Stat_mat (R1(*), R2(*), Zx(*), Zy(*), Stat(*), Cov(*), O_Set(*), N)
Print (Stat(*), O_Set(*), Cov(*), R(*), N, 2)
Ct_at (Zx(*), Zy(*), Ctt(*), Att(*), Wpt(Wt,5), Wpt(Wt,6), Wpt(Wf,5), Wpt(Wf,6), N, R, Angle)
Reg(Stat(*), Cov(*), S(*), R(*), O_set(*))

This program function converts Miniranger range data to cartesian xy coordinates relative to a local origin. Cross track and along track data are also calculated. Trilateration is used to calculate xy position using distances measured to two reference stations and the distance between reference stations. Cross track and along track position are calculated relative to a trackline between two waypoints.

Input parameters:

Rp$	- position reference file which contains name and xy position of reference stations and local grid origin.
Ref$(*)	- names of reference stations and local origin
Ref(*)	- xy position of reference stations and local origin
R1(*)	- range data to transponder R1
R2(*)	- range data to transponder R2
File$	- waypoint file name
F1	- indicator if waypoint file has been previously read
Wt	- waypoint to for along/cross track calculation
Wf	- waypoint from for along/cross track calculations
Wpt(*)	- waypoint data array.
Ctat$	- indicates if cross track and along track position is to be calculated. yes = "Y", no - "N"

Output parameters:

Zx(*), Zy(*) - position coordinates for each data sample
Att(*), Ctt(*) - along and cross track coordinates of each data sample - tt
Stat(*), Cov(*), S(*), R(*), O_Set(*) - statistics arrays which contain mean, standard deviation, covariance, correlation coefficients and linear regression parameters for range and position data (see K2).

Local Variables:

Zb(*)	- Local coordinates of the reference station at the B vertex of the triangle formed by the survey vessel and two reference stations
Zc(*)	- Local coordinates of the reference station at the C vertex of the triangle formed by the survey vessel and two reference stations.
R	- Range between reference stations (km)

Alpha - azimuth between reference stations

Point$ - location of R1 transponder
1 = B vertex
2 = C vertex

Point - Location of R1 transponder, B or C

Sign - Normally = 1; if reference station baseline is crossed, change to -1

Q - Sample number on which the sign is to be reversed

Rb - Side of triangle opposite B vertex (m)

Rc - Side of triangle opposite C vertex (m)

A,B,C - Angles of triangle formed by vessel and two reference stations, angles are labeled clockwise from vessel

User Instructions:

Prerequisite functions: K0 Read Data

1. Press K1. The menu is cleared and "CONVERT RANGE DATA TO XY POSITION" appears on the CRT in inverse video.

2. When "POSITION REFERENCE FILE NAME?" appears in display area (tape containing position reference file should be in right hand tape drive):

a. Enter: Position reference file name
b. Press: CONT

The file name is printed on the hard copy printer.

3. The title and position of the reference stations and the local grid origin and the range and bearing between reference stations are printed on the CRT. When "IS R1 AT POINT B(1) OR POINT C(2)? 1 or 2" appears in the display area:

a. If the R1 transponder was at Point B

(1) Enter: 1
(2) Press: CONT
(3) When "ENTER SIGN, +1 OR -1" appears, enter +1 if ship is above reference station base line, -1 if below
(4) When "DATA SAMPLE TO CHANGE SIGN" appears, enter the number of the data sample at which the survey vessel crosses the baseline

"R1 TRANSPONDER IS AT POINT B" and "CONVERTING DATA" (Blinking) will be printed on the CRT display.

b. If the R1 transponder was at point C:
(1) Enter: 2
(2) Press: CONT
(3) When "ENTER SIGN, +1 OR -1" appears, enter +1 if ship is above reference station base line, -1 if below

(4) When "DATA SAMPLE TO CHANGE SIGN" appears, enter the number of the data sample at which the survey vessel crosses the baseline

"R1 TRANSPONDER IS AT POINT C" and "CONVERTING DATA" (Blinking) will be printed on the CRT display.

4. The sample number and the angle A will appear on the display line. When the calculations are complete, a table of average ranges and xy position and their standard deviations is printed on the hard copy printer.

6. "Waypoint file?" Will appear on the display line if it had not been previously entered

a. Enter: waypoint file name (tape with waypoint file should be in right hand tape drive

b Press: CONT

6. When "WAYPOINT TO" appears on the display line:

a. Enter: Waypoint To
b. Press: CONT

7. When "WAYPOINT FROM" appears on the display line:

a. Enter: Waypoint from
b. Press: CONT

8. The program function is complete. The function menu is printed on the CRT.

Function: Calculate Statistics and Linear Regression of TD Data

Special Function Key: K2

Subprograms:

```
Hi_lo (W(*), N, Bw, Lw)
Stat_mat W(*), X(*), Y(*), Z(*), Stat(*), Cov(*), O_set(*), N)
Reg (Stat(*), Cov(*), S(*), R(*), O_set(*))
Print (Stat(*), O_set(*), Cov(*), R(*), N, V)
```

This program function calculates statistics, linear regression coefficients, and minimum and maximum for the four TD arrays. Statistics and regression parameters calculated are:

a. mean for each TD
b. standard deviation for each TD
c. covariance of TD pairs
d. correlation coefficient of each TD pair
e. linear regression slopes of each TD pair
f. standard deviation of residuals for each regression line

Input parameters:

W(*), X(*), Y(*), Z(*)	- TD arrays
N	- number of samples

Output parameters:

Stat(*)	- Summary statistics array containing sums, sums of squares, mean and standard deviation
O_set(*)	- First data sample for each TD
Cov(*)	- Summary statistics array contains sums of cross products, covariance, and correlation coefficients of TD pairs
Bw, Bx, By, Bz	- Maximum value for each TD array
Lw, Lx, Ly, Lz	- minimum value for each TD array.
S(*)	- A summary statistics array containing mean and standard deviation for each TD pair
R(*)	- An array containing linear regression slope, RMS residuals, and definition of the independent variable for each TD pair

<u>User Instructions:</u>

Prerequisite functions: K0 Read Data

1. Press K2. The menu is cleared and "STATS AND REGRESSION OF TD DATA" appears on the CRT in inverse video.

2. The file name, start time, stop time, and a tabulation of the statistics and linear regression parameters for the TDs and TD pairs is printed on the hard copy printer.

3. The program function is complete. The function menu is printed on the CRT.

Function: Compare Mini-ranger and Loran-C position data

Special Function Key: K3

Subprograms:

Compar (Pos_x(*), Pos_y(*), Zx(*), Zy(*), At(*), Ct(*), Att(*), Ctt(*), F$, N, R)

This program function compares Loran-C derived position with Mini-ranger calculated position. Mini-ranger positions are assumed to be "truth" and the difference between the Loran-C position and Mini-ranger position is the Loran-C position error. Two sets of position errors are calculated, Xy and along/cross track. The Xy errors are a direct comparison between Mini-ranger and Loran-C derived positions. The Mini-ranger along/cross track position is based on the local coordinates of the waypoints. The Loran-C along/cross track position is based on the difference in TDs between waypoints.

The following parameters are calculatred:

1. average x and y position error
2. rms x and y position error
3. average along and cross track error
4. rms along and cross track error
5. rms radial error based on xy position errors
6. rms radial error base on along/cross track errors

Input variables:

Pos_x(*)	- Loran-C derived x position
Pos_y(*)	- Loran-C derived y position
Zx(*)	- Mini-ranger derived x position
Zy(*)	- Mini-ranger derived y position
Ct(*)	- Loran-C derived cross track position
At(*)	- Loran-C derived along track position
Ctt(*)	- Miniranger derived cross track position
Att(*)	- Mini-ranger derived along track position
N	- Number of samples
F$	- Data file
R	- Range between waypoints

<u>User Instructions:</u>

Prerequisite functions: K0 - Read Data
K1 - Convert Range Data to xy Position
K17 - Convert TD data to xy Position

1. Press K3. The menu is cleared and "COMPARE MINI-RANGER AND LORAN-C POSITIONS" appears on the CRT.

2. A tabulation of Loran-C position error data will be printed on the hard copy printer.

3. A plot of x and y errors will appear on the CRT screen. When finished viewing the plot, press CONT.

4. When "HARD COPY" Y OR N" appears on the display line:

a. If a hard copy of the xy error plot is desired
   (1) Enter: Y
   (2) Press: CONT
b. If no hard copy is wanted:
   (1) Enter: N
   (2) Press: CONT

5. A plot of along and cross track errors will appear on the CRT screen. When finished viewing the plot, press CONT.

6. When "HARD COPY? Y OR N" appears on the display line:

a. If a hard copy of the xy error plot is desired:
   (1) Enter: Y
   (2) Press: CONT
b. If no hard copy is wanted:
   (1) Enter: N
   (2) Press: CONT

7. The program function is complete. The function menu is printed on the CRT.

Function: Plot TD Data with Regression Lines

Special Function Key: K4

Subprograms:

Plot (Lx, Bx, Ly, By, Xaxis, Yaxis, X(*), Y(*), Slope, N, "TDX", "TDY", F$)

The program function plots two of the time-difference arrays against each other and also plots the linear regression line for the TD pair chosen. The program automatically scales the plot for the minimum and maximum TDs for each array. The axes are drawn through the mean for each TD. Minor tic marks are every microsecond and major tic marks every 10 microseconds.

Parameters:

Plot - TD pair to be plotted

1 = WX; 2 = WY; 3 = WZ; 4 = XY, 5 = XZ, 6 = YZ

Lw, Lx, Ly, Lz - Minimum TD for each array

Bw, Bx, By, Bz - Maximum TD for each array

S(*) - Statistics summary array containing means and standard deviations for each pair to TD arrays

R(*) - Regression summary array which contains slopes of regression lines

N - Number of samples

F$ - File name

<u>User Instructions</u>:

Prerequisite function: K0 - Read Data File
K2 - Calculate Statistics and Linear Regression of TD Data

1. Press K4. The menu is cleared and "PLOT TD DATA WITH REGRESSION LINES" appears on the CRT in inverse video.

2. When "PLOT? WX1, WY2, WZ3, XY4, XZ5, YZ6" appears on the display line:

   a. Enter: 1-6 depending on which plot is desired
   b. Press: CONT

3. The selected TD data will be plotted on the CRT. When finished viewing the plot:

   a. Press: CONT

4. When "HARD COPY? Y OR N" appears on the display line:

   a. If a hard copy of the plot is desired:
      (1) Enter: Y
      (2) Press: CONT
   b. If a hard copy is not desired:
      (1) Enter: N
      (2) Press: CONT

5. The program function is complete. The function menu is printed on the CRT.

Function: Plot Residuals

Special Function Key: K5

Subprograms: Rplot(Iv(*), D(*), S(*), R(*), Pr, N, V1$, V2$, F$)
Hi_lo (X(*), N, Hi, Lo)

This program function plots the residuals from the linear regression of any TD pair (See K2). The residuals may be plotted against sample number or the independent variable. Residuals are normalized to the standard deviation of the residuals. Normalized values greater than 5 are printed on the hard copy printer and are not plotted.

Variables:

W(*), X(*), Y(*), Z(*) - TD data arrays
S(*) - Statistics summary array
R(*) - Regression summary array
Pr - TD pair for which residuals are being plotted: 1 = WX, 2 = WY, 3 = WZ, 4 = XY, 5 = XZ; 6 = YZ
N - Numbers of Samples
V1$, V2$ - Names of two TDs for which residuals are being plotted, i.e. "TDW", "TDX", etc.
F$ - Data file name

<u>User Instructions</u>:

Prerequisite functions:
- K0 - Read Data
- K2 - Calculate Statistics and Linear Regression of TD Data

1. Press K5. The menu is cleared and "PLOT RESIDUALS" is printed on the CRT in inverse video.

2. When "PLOT? WX1, WY2, WZ3, XY4, XZ5, YZ6" appears on the display line:

   a. Enter: 1-6 depending on which plot is desired
   b. Press: CONT

3. When "PLOT RESIDUALS VS N(1) OR INDEPENDENT VAR(2)?" appears on the display line:

   a. To plot the residuals vs sample number:
      (1) Enter: 1
      (2) Press: CONT
      (3) Go to: Step 4
   b. To plot the residuals vs the independent variable:
      (1) Enter: 2
      (2) Press: CONT
      (3) Go to: Step 4

4. The normalized residuals for the selected TD pair regression line will be plotted on the CRT. When finished viewing the plot:

   a. Press: CONT

5. When "HARD COPY? Y OR N" appears on the display line:

   a. If a hard copy of the plot is desired:
      (1) Enter: Y
      (2) Press: CONT
   b. If no hard copy of the plot is desired:
      (1) Enter: N
      (2) Press: CONT

6. The program function is complete. The function menu is printed on the CRT.

Function: Edit Data

Special Function Key: K16

Subprograms:

Delete (W(*)), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
Delete_blok (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
Delete_td (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
Delete_rg (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
Store(W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)

This program function is used to edit (i.e. remove) samples from the data arrays. Four options are available to the user:

a. The first option deletes a single data sample in each data array.
b. The second deletes a block of data samples.
c. The third deletes samples with TD samples outside a range that is input by the operator.
d. The fourth deletes samples with outliers in the range data. The routine compares each range with the average of the previous and next sample. If there is a difference of more than 100 meters, the sample is deleted. Unfortunately, this technique usually deletes the outliers and the sample on each side of the outlier. This is normally not a significant problem, i.e. nine samples will be deleted rather than three. The alternative is to detect the outliers and the corresponding sample number using the Plot XY Position function (K18) and use the single sample delete option.

When editing of the data arrays is complete, the operator may store the edited data on a new file for later use.

NOTE: The edit function deletes only data samples. Arrays containing XY and along/cross track position data are not affected. Functions K1 and K17 must be repeated to reflect the result of this editing on the computed positions.

Paramaters:

Edit - Indicates which edit subroutine is to be used.

1 - delete sample by sample
2 - delete block of samples
3 - delete samples with TD values outside limits determined by operator
4 - delete samples with range outliers

W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*) - Data arrays
N - Number of samples
F$ - Data file name

User Instructions:

Prerequisite functions:

K0: Read Data
Optional: Several other program functions may be run prior to the Edit function to determine which data samples are to be edited, e.g. K2, K4, K5, K1, K18

1. Press: K16. The menu is cleared and "EDIT DATA" appears on the CRT in inverse video.

2. When "SINGLE LINE1, BLOCK2, TD-CLIP3, OR RANGE4?" appears on the display line:

a. If you want to delete samples one at a time:
   (1) Enter: 1
   (2) Press: CONT
   (3) Go to Step 3
b. If you want to delete a block(s) of samples:
   (1) Enter: 2
   (2) Press: CONT
   (3) Go to Step 8
c. If you want to delete samples outside a set of TD bounds:
   (1) Enter: 3
   (2) Press: CONT
   (3) Go to Step 11
d. If you want to delete outliers in the range data:
   (1) Enter: 4
   (2) Press: Cont
   (3) Go to Step 15

e. If you want to continue without any editing:
   (1) Press: CONT without entering any number
   (2) Go to step 5

3. When "SAMPLE TO BE DELETED? START WITH HIGHEST NUMBER" appears on the display line:

a. Enter: Sample number
b. Press: CONT

The sample deleted and the number of samples remaining are printed on the hard copy printer.

4. When "ANOTHER SAMPLE TO DELETE?" appears in the display area:

a. Enter: Y(if another sample is to be deleted) or N
b. Press: CONT
c. If another sample is to be deleted, go to Step 3.

5. When "EDIT MORE DATA? Y or N" appears in the display area:

a. if more data editing is desired:
   (1) Enter: Y
   (2) Press: CONT
   (3) Go to step 2

b. if no more editing is to be performed:
   (1) Enter: N
   (2) Press: CONT
   (3) Go to step 6

6. When "LIST DATA?" appears in the display area:

a. If a listing of the data is desired:
   (1) Enter: Y
   (2) Press: CONT. The data is printed on the hard copy printer.
b. If a listing of the data is not desired:
   (1) Enter: N
   (2) Press: CONT

7. When "STORE EDITED DATA?" appears in the display area:
a. If you want to store the data
   (1) Enter: Y
   (2) Press: CONT
   (3) Go to Step 8
b. If you do not want to store the data
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 16

8. Insure that a tape cartridge is loaded into the left hand tape drive. When "FILE NAME?" appears on the display area:
a. Enter: File name
b. Press: CONT
c. Go to Step 16

9. When "FIRST SAMPLE IN BLOCK?" appears in the display area (if more than one block is to be deleted, delete the block with the higher sample numbers first. The array is renumbered after each edit.)

a. Enter: First sample
b. Press: CONT

10. When "LAST SAMPLE IN BLOCK TO BE DELETED" appears in the display area:

a. Enter: Last sample
b. Press: CONT

The block of samples deleted and the numbers of samples remaining are printed on the hard copy printer.

11. When "ANOTHER BLOCK TO BE DELETED?" appears in the display area

a. If there is more data to be deleted
   (1) Enter: Y
   (2) Press CONT
   (3) Go to Step 9
b. If no more data is to be deleted
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 5

12. When "CLIP LIMITS FOR WHICH SECONDARY? W1, X2, Y3, Z4" appears in the display area:
a. Enter: 1 through 4 to select the secondary to test for clip
b. Press: CONT

13. "SETTING CLIP LIMITS FOR ( )" will be printed and "CLIP LIMITS? MIN,MAX" will appear in the display area:

a. Enter: minimum TD, maximum TD
b. Press: CONT

The number of samples deleted and number of samples remaining are printed on the hard copy printer.

14. When "SET ADDITIONAL CLIP LIMITS?" appears in the display area:

a. If more clip limits are to be set:
    (1) Enter: Y
    (2) Press: CONT
    (3) Go to step 12

b. If no more clipping is to be performed:
    (1) Enter: N
    (2) Press: CONT
    (3) Go to step 5

15. The program will delete what it determines to be range outliers. Generally an outlier and the sample before and after it will be deleted. The samples deleted, the total number of samples deleted, and the remaining number of samples is printed on the hard copy printer.

a. Go to Step 6

16. The program function is complete. The program menu is printed on the CRT.

Function: Convert TD Data to XY Position

Special Function Key: K17

Subprograms:

Cart_coord (Xmit(*), Wpt(25,7), Wpt(25,8), Zxmit(*))
Rb (Wpt(Wt,5), Wpt(Wt,6), Zmit(I,1), Zxmit(I,2),Bear(I), Range (I))
G_mat (Power(*), V, Range(*), Bear(*), Zp(*), Aa(*), G123(*), G12(*), G23(*), G13(*))
Wp3 (Wpt(*), Conf, Wt, Tp(*), Zpp(*), L$)
Td3 (W(*), X(*), Y(*), Z(*), I, Conf, Tq(*))
Fehg (Zxmit(*), Zpp(*), Tp(*), Tq(*), G123(*), Zq(*), V, Nsta)
Wp2 (Wpt(*), Conf, Wt, Tpp(*), Pair, Zpp(*), G12(*), G23(*), G13(*), G(*), L$, Zxmit(*), Zxm(*))
Td2 (W(*), X(*), Y(*), Z(*), I, Conf, Pair, Tqq(*))
Wpf2 (Wpt(*), Conf, Pair, Wf, Tqq(*)
Ct_at (Pos_x(*), Pos_y(*), Ct(*), At(*), Wpt(Wt,5), Wpt(Wt,6), Zq(1,1), Zq(2,1), N, R, Angle)
Stat_mat (Pos_x(*), Pos_y(*), Ct(*), At(*), Stat(*), Cov(*), O_set(*), N)
Reg (Stat(*), Cov(*), S(*), R(*), O_set(*))
Track (Stat(*), O_set(*), Cov(*), R(*), Sample, F$, Wt, Wf, Angle, N)
Rotate (Glat, Glon, R(*))
Cart (Plat, Plon, X, Y)
FNGlat (Lat, F)
Pseudo (Glat, Glon, R*, Plat, Plon)

This program function converts Td data to xy and along/cross track positions using the Flat Earth Hyperbolic Grid (FEHG) algorithm and surveyed waypoints. The program will compute either a three or two TD solution. A summary table is printed on the hard copy printer which lists:

-chain and LOPs used in the solution
-file name
-bearing angle between waypoints used for along/cross track calculation
- rms trackline of data
-average cross track position
-standard deviation of cross track position
-average xy position
-standard deviation of xy position data

Function Parameters:

Chain$ - Loran-C transmitting chain file which contains transmitter positions, power, and emission delays. The convention for naming chain files is a four letter code followed by a number from 1 to 4, eg NEUS1. The number designates the three secondaries utilized: 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ.
C1 - Flag which is set when the Loran-C chain file is read
Pos_x(*), Pos_y(*) - x and y position arrays
At(*), Ct(*) - Along and cross track position arrays
Xmit(*) - Transmitter geodetic positions
Power(*) - Transmitter power levels
Emis(*) - Secondary transmitter emission delays
Fl - Flag to indicate when the waypoint file is read

File$ - Waypoint file name
Wpt(*) - Waypoint data array
Zxmit(*) - Transmitter xy positions referenced to local origin (Wpt(25,7), Wpt(25,8))
Wt - Waypoint used as reference for TD to Xy calculations
Wf - Waypoint used for along/cross track calculations to Wt
Bear(*) - Bearings from waypoint to transmitters
Range(*) - Ranges from waypoint to transmitters
Zp(*) - XY position of waypoint
V - Velocity of propagation
As(*) - Gradient matrix, xy to TD
G123(*) - Three Td gradient matrix, TD to XY
G12(*), G13(*), G23(*) - Two TD gradient matrices, TD to XY
Conf - Chain configuration 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ
Ch$ - Abbreviation of chain, eg first four characters of chain$
So - Indicates two or three solution: 2 = 2-TD, 3 = 3-TD
L$ - LOPs used in TD to XY calculation, eg XYZ, YZ, WX, etc.
W(*), X(*), Y(*), Z(*) - TD data arrays
Zq(*) - XY position solution
Tq(*) - 3-TD sample
TQQ(*) - 2-TD sample
Zpp(*) - XY position of waypoint
Tp(*) - 3-TD waypoint
Tpp(*) - 2-TD waypoint
Zxx(*) - Dummy position array
Pair - LOP pair used for TD to xy calculation
  If Conf = 1; 1 = XY, 2 = XZ, 3 = YZ
  If Conf = 2; 1 = WX, 2 = WY, 3 = XY
  If Conf = 3; 1 = WX, 2 = WZ, 3 = YZ
  If Conf = 4; 1 = WY, 2 = WZ, 3 = YZ
G(*) - 2TD G matrix used in 2TD solution
Zxmit(*) - Transmitter xy positions
R - Distance between "TO" and "FROM" waypoints
Stat(*), Cov(*), O_set(*), S(*), R(*) - Arrays containing summary statistics of xy and along/cross track position
Nsta - Number of secondary stations used in the FEHG computations

User Instruction:

Prerequisite instructions: Read Data, KO

1. Press K17. The menu will be cleared and "CONVERT TD TO XY" will appear on the CRT in inverse video.

2. When "CHAIN CONFIGURATION? " appears on the display line:

(1) Insure that the tape cartridge containing chain data is in the right hand tape drive.
(2) Enter: Chain file. Note: Chain files have five character names. The first four characters denote the chain, eg NEUS, SEUS, WCUS. The number following the characters denotes the secondaries, eg: 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ
(3) Press: CONT

NOTE: Chain configuration data is only read once. The data is stored in memory for further use. To change chain configuration either:

a. Press: STOP
b. Press: RUN. This clears data memory and "CHAIN CONFIGURATION" will be requested as above. All other data in memory will also be cleared.

OR

a. Enter: Cl = 0
b. Press: EXECUTE. This clears the flag, Cl, which will cause the program to branch through the "CHAIN CONFIGURATION?" statement.

3. When "WAYPOINT FILE NAME?" appears in the display area:

a. Insure that the tape cartridge containing waypoint data is in the right hand tape drive.
b. Enter: Waypoint file
c. Press: CONT

NOTE: Waypoint file data is read only once. The data is stored in memory for further use.

To change the waypoint file:

a. Press: STOP
b. Press: RUN. This clears data memory and "WAYPOINT FILE" will be requested as above

OR

a. Enter: Fl = 0
b. Press: EXECUTE. This clears the flag Fl which will cause the program to branch through the "WAYPOINT FILE?" statement.

4. When "WAYPOINT TO?" appears in the display area:

a. Enter: Waypoint number. NOTE: If a comparison of Mini-ranger and Loran-C position data is to be made, insure that "TO" and "FROM" waypoints are consistent. Conversion waypoint is "TO" waypoint.
b. Press: CONT

5. When "WAYPOINT FROM?" appears in the display area:

a. Enter: Waypoint number. See Note above
b. Press: CONT

6. When "TWO OR THREE SOLUTION?" appears in the display area:

a. If a two TD solution is desired:
   (1) Enter: 2
   (2) Press: CONT
   (3) "TWO TD SOLUTION, CHAIN =   LOPS =   " is printed on the hard copy printer.
b. If a three TD solution is desired:
   (1) Enter: 3
   (2) Press: CONT
   (3) "THREE TD SOLUTION, CHAIN =   LOPS =   " is printed on the hard copy printer.

7. The sample number is displayed in display area as the program sequences through the data arrays. When the calculations are complete, a summary table is printed on the hard copy printer which contains:

a. data file name
b. bearing angle between waypoints
c. rms trackline of data
d. average cross track position
e. standard deviation of cross track position
f. average x,y position
g. standard deviation of xy position data

8. The program function is complete. The function menu is printed on the CRT.

Function: Plot XY Data

Special Function Key: K18

Subprograms:

Hi_lo (Pos_x(*), N, Maxx, Minx)
Xplot (T, Pos_x(*), Pos_y(*), Minx, Maxx, Miny, Max, y, N, No, F$, Ch$, L$, Pl)
X_errplot (1, Zx(*), Zy(*), Pos_x(*), Pos_y(*), Minx, Maxx, Miny, Maxy, N)

The program function plots the xy data calculated from Loran-C TDs and/or Mini-ranger data. Four options are available:

1. Mini-ranger data only
2. Loran-C data only
3. Mini-ranger and Loran-C data
4. Error plot

The first two options allow the operator to blow-up a section of the plot, to find the sample numbers of outliers, and to digitize up to 10 locations on the plot. A hard copy option is provided with all four plots. The "Error plot" plots a vector at each sample. The tail of the vector is the mini-ranger position fix; the head (denoted by a "o") is the Loran-C position fix.

Function parameters:

Pos_x(*), Pos_Y(*) - xy position calculated from Loran-C data
Zx(*), Zy(*) - xy position calculated from Mini-ranger data
Minx, Miny - Minimum x and y positions
Maxx, Maxy - Maximum x and y positions
T - Line type
N - Number of samples
No - Denotes first or second time data is being plotted
Ch$ - Loran-C chain, eg NEUS, SEUS, etc.
L$ - Loran-C LOPs, eg XY, WX, etc.
Pl - Plot: 1 = Mini-ranger, 2 = Loran-C, 3 = Mini-ranger and Loran-C, 4 = Error Plot

User Instructions:

Prerequisite functions: K0, K1, and K17

1. Press K18: The menu is cleared and "PLOT XY DATA" is printed on the CRT.

2. When "MINI-RANGER 1, CONVERTED LORAN-C2, BOTH3, OR ERROR PLOT4?" appears on the display line:

a. For a plot of Mini-ranger fixes:
   (1) Enter: 1
   (2) Press: CONT
   (3) Go to Step 3

b. For a plot of Loran-C fixes:
(1) Enter: 2
(2) Press: CONT
(3) Go to Step 3

c. For a plot of both Loran-C and Mini-ranger fixes:
(1) Enter: 3
(2) Press: CONT
(3) Go to Step 11

d. For an error plot:
(1) Enter: 4
(2) Press: CONT
(3) Go to Step 12

3. The xy data is plotted on the CRT. After viewing the plot:

a. Press: CONT

4. When "ZOOM: Y OR N" appears on the display line:

a. To blow up a section of the plot:
(1) Enter: Y
(2) Press: CONT

b. If a "zoom" or further "zoom" is not wanted
(1) Enter: N
(2) Press: CONT
(3) Go to Step 6

5. When the cursor appears on the CRT, place it at the lower left-hand corner of the area of the interest using the four directional arrow keys and press CONT. When the cursor reappears, position it at the upper right hand corner of the area of interest and press CONT. The selected area is replotted on the CRT.

Go to Step 3

NOTE: The "zoom" may be repeated several times.

6. When "FIND SAMPLE NUMBER AND VALUE OF PLOTTED DATA POINT?" appears on the display line:

a. If the sample number of a point on the plot is desired:
(1) Enter: Y
(2) Press: CONT

b. If not:
(1) Enter: N
(2) Press: CONT
(3) Go to Step 8

7. When the cursor appears on the CRT:

a. Center it on the point of interest
b. Press: CONT

The sample number and coordinates will be printed on the CRT. If no data is printed, try repositioning the cursor closer to the point of interest or replot the data and use "zoom" to blow up the area around the point of interest.

c. Go to Step 6

8. When "DIGITIZE?" appears in the display area:

   a. If you want to digitize 1-10 points on the plot:
      (1) Enter: Y
      (2) Press: CONT
   b. If not:
      (1) Enter: N
      (2) Press: CONT
      (3) Go to Step 10

9. When "HOW MANY? 10" appears in the display area:

   a. Enter: the number of points to be digitized
   b. Press: CONT

When the cursor appears on the CRT, center it over a point of interest and

   c. Press: CONT

A plus sign (+) will appear over the data point and the number of points digitized (1-10) will appear to its right. Continue the procedure until the number of points to be digitized is complete. A list of the digitized point coordinates is printed on the hard copy printer.

   d. Go to Step 13

10. The plot will reappear on the CRT. When finished viewing the plot press CONT.

11. The Loran-C and Mini-ranger data converted to xy position is plotted on the CRT. The Mini-ranger data is the solid line. The Loran-C data is the dotted line.

   Go to Step 13

12. The Loran-C and Mini-ranger xy data is plotted on the CRT in the form of a vector. The tail of the vector is the Mini-ranger position, the head is the Loran-C position which is marked with an "o."

13. When "HARDCOPY? Y OR N" appears on the display line:

   a. If a hard copy of the plot is desired:
      (1) Enter: Y
      (2) Press: CONT
   b. If not:
      (1) Enter: N
      (2) Press: CONT

14. The function is complete. The function menu is printed on the CRT.

Function: Plot Along/Cross Track Data

Special Function Key: K19

Subprograms:

Hi_lo (Ct(*), N, Maxc, Minc)
Patct (At(*), Ct(*), N, Mina, Maxa, Minc, Maxc, Wt, Wf, F$, R, No, Data)

This program function plots cross track vs along track position. Three options are available:

1. Mini-ranger data
2. Loran-C data
3. Mini-ranger and Loran-C data

The plots are automatically scaled and labeled.

Function parameters:

At(*), Ct(*) - Along and cross track position arrays calculated from Loran-C data
Att(*), Ctt(*) - Along and cross track position array calculated from Mini-ranger data
N - Numbers of samples
Maxc, Maxcc - Maximum cross track distance
Minc, Mincc - Minimum cross track distance
Maxa, Maxaa - Maximum along track distance
Mina, Minaa - Minimum along track distance
Wt - Waypoint to
Wf - Waypoint from
R - Distance between waypoints
F$ - File name
Data - Plot option: 1 = Mini-Ranger, 2 = Loran-C, 3 = Mini-Ranger and Loran-C

<u>User Instructions:</u>

Prerequisite function: K1 and/or K17

1. Press K19. The program menu is cleared and "PLOT ALONG TRACK CROSS TRACK DATA" is printed on the CRT in inverse video.

2. When "MINIRANGER1, LORAN-C2, OR BOTH3? 1, 2 OR 3" appears in the display area:

a. If a plot of the along/cross track positions computed from Mini-ranger data is desired:
(1) Enter: 1
(2) Press: CONT
(3) Go to Step 3

b. If a plot of along/cross track positions computed from Loran-C data is desired:

(1) Enter: 2
(2) Press: CONT
(3) Go to Step 3

c. If a plot of cross/along track positions computed from both Mini-ranger and Loran-C is desired:

(1) Enter: 3
(2) Press: CONT
(3) Go to Step 3

3. The along/cross track data selected is plotted on the CRT. When finished viewing the plot:

a. Press: CONT

4. When "HARD COPY? Y OR N" appears in the display area:

a. If a hard copy of the plot is desired:

(1) Enter: Y
(2) Press: CONT

b. If not:

(1) Enter: N
(2) Press: CONT

5. This function is complete. The function menu is printed on the CRT.

Function: Predict TD

Special Function Key: K20

Subprograms:

Range (Lat, Lon, Xmit(I,1), Xmit(I,2), Range (I))
Bear (Lat, Lon, Xmit(I,1), Xmit(I,2), Bear(I))
Td (Range(*), V, Emis(*), Td(*))
G_mat(Power(*), V, Range(*), Bear(*), Zp(*), Aa(*), G123(*), G12(*), G23(*), G13(*))

This program function calculates predicted TDs for an input latitude, longitude. The program also outputs range and bearing to transmitter stations and GDOPs for the three TD and two TD fixes.

Function Variables:

Cl - Chain flag indicates if chain data has been read into memory
Chain$ - Chain name
Xmit(*) - Transmitter positions (latitude, longitude)
Power(*) - Transmitter power
Emis(*) - Secondary emission delays
P$ - Indicates input position is a waypoint
Fl - Waypoint table flag, indicates waypoint table has been read into memory
Wpt(*) - Waypoint table
Lat,Lon - Latitude and Longitude of position of interest
D,M,S - Degrees, Minutes, Seconds
Range (*) - Ranges to transmitters
Bear(*) - Bearings to transmitters
Td(*) - Predicted TDs
Aa(*) - Gradient matrix; xy to TD
G123(*) - Three TD gradient matrix; TD to XY
G12(*), G13(*), G23(*) - Two TD gradient matrices, TD to XY
Gdop 123 - Three TD Geometric Dilutin of Precision (GDOP)
Gdop12, Gdop13, Gdop23 - Two TD GDOPs
Ch$ - Chain abbreviation, eq NEUS = North East United States
Conf - Indicates three TD configuration: 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ

User Instructions:

Prerequisite functions: None

1. Press K20. The program menu is cleared and "TD PREDICTION" is printed on the CRT in inverse video.

2. When "CHAIN FILE?" appears in the display area:

(1) Insure that the tape cartridge containing chain data is in the right hand tape drive

(2) Enter: Chain file. Note: Chain files have five character names. The first four characters denote the chain, eg NEUS SEUS, etc. The number following the character denotes the secondaries, eg 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ

(3) Press: CONT

Note: Chain configuration data is read only once. The data is stored in memory for further use. To change chain configuration either:

a. Press: STOP

b. Press: RUN. This clears data memory and "CHAIN FILE" will be requested as above. All other data in memory will also be cleared

OR

a. Enter: C1 = 0

b. Press: EXECUTE. This clears the flag C1 which will cause the program to branch through the "CHAIN FILE" statement.

3. When "IS POINT OF INTEREST ON WAYPOINT FILE? Y OR N" appears in the display area:

a. If the point of interest is on the waypoint file:
   (1) Enter: Y
   (2) Press: CONT

b. If not:
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 6

4. When "WP FILE NAME?" appears on the display line (see Note below)

a. Insure that the tape cartridge containing waypoint data is in the right hand tape drive

b. Enter: Waypoint file name

c. Press: CONT

Note: This step is skipped on successive runs of this program function or if the waypoint file has been read during a previous function.

5. When "WAYPOINT NO.?" appears on the display line:

a. Enter: the waypoint of interest

b. Press: CONT

c. Go to Step 8

6. When "INPUT LAT IN DEGREES, MIN, SEC" appears on the display line:

a. Enter: Latitude of position of interest (Degrees, Minutes, Seconds). The program assumes North latitude

b. Press: CONT

7. When "INPUT LON IN DEGREES, MIN, SEC" appears on the display line:

a. Enter: Longitude of position of interest (Degrees, Minutes, Seconds). The program assumes West longitude

b. Press: CONT

8. The program calculates and prints the following parameters:

   a. Predicted time differences. (If a waypoint is selected, surveyed TDs are also listed.)
   b. Ranges and bearings to transmitters
   c. GDOPs for the three TD and each of the two TD combinations.

9. The program function is complete. The program menu is printed on the CRT.

Function: Calculate TD Grid Warp

Special Function Key: K21

Subprograms:

```
Cart_coord (Xmit(*), Wpt (25,7), Wpt (25,8), Zxmit(*))
Warp (X(*), Y(*), Z(*), Zx(*), Zy(*), Zp(*), T(*), Zxmit(*), V, N, Att(*),
  R, Wt, Wf, Conf, F$)
T_errplot(A(*), B(*), C(*), At(*), R, N, Conf, Wt, Wf, F$)
```

This program function calculates the difference between measured TDs and TDs projected from a nearby waypoint. This difference is termed "TD grid warp" and is basically a measure of the change in Additional Secondary Phase Factor (ASF) over the area where data was collected. TDs are projected from the waypoint to the measurement point based on the difference between the measured and waypoint positions. There is a small error in the calculation due to use of a flat earth model for transmitter locations. This error is typically less than 40 nanosec within 12KM of the waypoint.

Function parameters:

W(*), X(*), Y(*), Z(*) - TD data arrays
T(*) - Waypoint TDs
Xmit(*) - Transmitter geodetic coordinates
Zxmit(*) - Transmitter xy position referenced to local origin
Wpt(*) - Waypoint table
Zp(*) - Waypoint position
V - Velocity of propagation
Zx(*), Zy(*) - Measured position arrays
N - Number of samples
Att(*) - Along track distance array
R - Range between waypoints
Wt, Wf - Waypoint to and Waypoint from
Conf - Chain configuration; 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ
F$ - Data file name

User Instructions:

Prerequisite functions:
- (1) Read Data; K0
- (2) Edit Data; K16 (optional)
- (3) Convert Range Data to XY Position; K1

1. Press K21. The menu is cleared and "CALCULATE GRID WARP" appears on the CRT in inverse video.

2. If "CHAIN FILE?" appears in the display line:

   (1) Enter: Five character chain neumonic (eg NEUS1, GILK1, etc.). Insure that the tape cartridge containing the waypoint file is in the right hand tape drive.
   (1) Press: CONT

3. The program function uses the "waypoint to" used for along/cross track calculations in function K1 as the reference waypoint. The mean and standard deviation of the TD errors for each TD is printed on the hard copy printer. A plot of the smoothed TD errors for each TD is plotted on the CRT. After viewing the plot:

a. Press: CONT

4. When "HARD COPY? Y OR N" appears on the display line:

a. If a hard copy of the plot is desired:
   (1) Enter: Y
   (2) Press: CONT
b. If not
   (1) Enter: N
   (2) Press: CONT

5. The function is complete. The CRT is cleared and the function menu is printed on the CRT.

Function: Daisy Chain

Special Function Key: K22

Subprograms:

Cart_coord (Xmit(*), Wpt(25,7), Wpt(25,8), Zxmit(*))
Rb(Wpt(Rw,5), Wpt(Rw,6), Zxmit(I,1), Zxmit(I,2), Bear (I), Range (I))
G_mat (Power(*), V, Range(*), Bear(*), Zp(*), Aa(*)), G123(*), G12(*), G23(*), G13(*))
Wp2 (Wpt(*), Conf, Rw, Tpp(*) Pair, Zpp(*), G12(*), G23(*), G13(*), G(*), L$,Zxmit(*), Zxm(*))
Wpf2 (Wpt(*), Conf, Pair, Iw, Tqq(*))
Wp3 (Wpt(*), Conf, Rw, Tp(*), Zpp(*), L$)
Fehg (Zxmit(*), Zpp(*), Tp(*), Tq(*), G123(*), Zq(*), V, Nsta)
Rotate (Glat, Glon, R(*))
Cart (Plat, Plon, X, Y)
FNGlat (Lat, F)
Pseudo (Glat, Glon, R*, Plat, Plon)

This program function calculates the position of a waypoint based on the difference in TDs between it and a neighboring waypoint. The Flat Earth Hyperbolic Grid (FEHG) is used to calculate position coordinates. Differential xy coordinates, differential latitude and longitude, and range and bearing between waypoints is also calculated. The user has the option of inserting the calculated xy coordinates (and latitude longitude) into the waypoint table, Wpt(*). Note: This function does not restore the new waypoint table in magnetic tape. If this is desired, function K23 must be used.

Function variables:

C1 - Status variable which indicates if chain data has been read into memory
F1 - Status variable which indicates if the waypoint table has been read into memory
Chain$ - Chain file
Xmit(*), Power(*), Emis(*) - Chain variables; geodetic coordinates, power levels, and emission delays
Conf - Chain configuration
Ch$ - First four characters of Chain$
Wpt(*) - Waypoint table
Zxmit(*) - Transmitter xy coordinates referenced to local origin (three TD case)
Rw - Reference waypoint (start)
Iw - Waypoint of interest (stop)
Bear(*), Range(*) - Bearings and ranges to transmitters
Zpp(*), Zp(*) - Reference waypoint position
V - Velocity of propagation
Aa(*), G123(*), G12(*), G13(*), G23(*) - gradient matrices
Zxm(*) - Transmitter xy coordinates referenced to local origin (two TD case)
So - Indicates two or three TD solution of FEHG
Tpp(*), Tp(*) - Two and three TD waypoints (start)

Tqq(*), Tq(*) - Two and three TD waypoints (stop)
Nsta - Number of secondary stations used in FEHG computations
Pair - TD pair (two TD case)
G(*) - Gradient matrix associated with TD pair (Pair)
L$ - Lops e.q. "XYZ," "XY," "XZ," etc.
Zq(*) - Waypoint of interest x, y position
Zxx(*) - Dummy position array
Dx, Dy - Difference in xy position of waypoints
Dlat, Dlon - Difference in latitude, longitude of waypoints
Lat, Lon - Computed latitude, longitude of waypoint
A - Angle between waypoints
R - Range between wayopints

User Instructions:

Prerequisite functions: assumes a waypoint file has been created and it contains TDs for the two waypoints and xy positions for the reference waypoint.

1. Press K22. The program menu is cleared and "DAISY CHAIN" appears on the CRT in inverse video.

2. If "CHAIN FILE?" appears in the display line:

   (1) Enter: Five character chain file (e.g. NEUS2, GTLK1, etc.) Insure that the tape cartridge containing the file is in the right hand tape drive.
   (2) Press: CONT

3. If "WP FILE?" appears in the display area:

   a. Enter: Waypoint file name. Insure that the tape cartridge containing the waypoint file is in the right hand tape drive.
   Press: CONT

4. When "INPUT START WAYPOINT NO." appears in the display area:

   a. Enter: Start or reference waypoint
   b. Press: CONT

5. When "END WAYPOINT" appears in the display area:

   a. Enter: Waypoint of interest
   b. Press: CONT

6. When "TWO OR THREE TD SOLUTION, 2 OR 3" appears in the display area:

   a. If a three TD solution is desired:
      (1) Enter: 3
      (2) Press: CONT
      (3) Go to Step 8
   b. If a two TD solution is desired
      (1) Enter: 2
      (2) Press: CONT

7. When "INPUT TD PAIR; XY1, XZ2, XY3" appears in the display area: (The choice of TD pairs depends on the chain configuration, e.g. if Conf = 2, the choices are WX, WY, XY)

a. Enter: Number corresponding to TD pair desired
b. Press: CONT

8. A table is printed on the hard copy printer which contains the following information:

a. the chain and lops used to calculate the waypoint position
b. the number of the waypoint used as the origin of the local grid
c. the two waypoints numbers
d. the differential xy positions and differential latitude, longitude
e. the angle and range between waypoints
f. the projected waypoint position coordinates based on the waypoint TDs
g. the current values of the waypoint position coordinates

9. When "CHANGE POSITION AND LAT/LON OR WAYPOINT OF INTEREST? Y OR N" appears on the display line:

a. If you want to change the values
   (1) Enter: Y
   (2) Press: CONT. Note: the values will be changes in the waypoint table, but not on magnetic tape. Function K23 must be used to restore the table on tape.
   (3) "WAYPOINT TABLE CHANGED" is printed on the hard copy printer.

10. The program function is complete. The CRT is cleared and the program menu is printed.

Function: Waypoint File

Special Function Key: K23

Subprograms:

Wpfile (Wpt(*), Fl, File$)

This program function is used to create waypoint table files and to edit and restore waypoint table files. It can also be used to obtain a listing of the waypoint table. The waypoint table is a 25X8 matrix. The first four columns contain waypoint TD values (TDW, TDX, TDY, TDZ). Columns five and six contain the xy position referenced to the local origin. Columns seven and eight contain the latitude and longitude in decimal degrees. Waypoint 25 (e.g. row 25) is designated as the local origin. The latitude and longitude stored in this location is used by any program function which calculates the local xy coordinates of the transmitters. If the local origin is also one of a sequence of waypoints, its parameters will be stored twice, i.e. in row 25 and in the row corresponding to its logical waypoint number.

Function Variables:

Wpt(*) - Waypoint table
File$ - Waypoint file name
Fl - Indicates if the waypoint file name has been previously entered

User Instructions:

Prerequisite functions: None

1. Press K23. The program menu is cleared and "FILE WAYPOINT DATA" appears on the CRT in inverse video.

2. If "WAYPOINT FILE NAME?" appears on the display line:

a. Enter: File name (6 characters or less)
b. Press: CONT

When "DOES THIS FILE CURRENTLY EXIST ON TAPE? Y OR N" appears in the display area:

c. Insure the tape cartridge which contains or is to contain the waypoint table is in the right hand tape drive.
d. If the file currently exists:
(1) Enter: Y
(2) Press: CONT
e. If not:
(1) Enter: N
(2) Press: CONT

3. When "WAYPOINT NUMBER?" appears in the display area:

   a. Enter: Waypoint number
   b. Press: CONT

4. When "INPUT OR CHANGE TDs? Y OR N?" appears in the display area:

   a. If you want to input or change TDs OR if you want to observe the TDs currently in the table:
      (1) Enter: Y
      (2) Press: CONT
      (3) Go to Step 5
   b. If not:
      (1) Enter: N
      (2) Press: CONT
      (3) Go to Step 6

5. When "TDW, TDX, TDY, TDZ?" appears on the display line:

   a. If you want to observe what values are currently stored for the waypoint selected
      (1) Press: CONT
          The current values will be printed on the hard copy printer.
   b. If you want to enter new values or change an old value:
      (1) Enter: Td values for TDW, TDX, TDY, and TDZ separated by commas. All TDs must be entered. If a value is unknown (or a don't care), enter a 0 in the proper location.
      (2) Press: CONT
      The TD values entered will be printed on the hard copy printer. Errors can be corrected in step 10.

6. When "INPUT OR CHANGE XY POSITION? Y OR N" appears on the display line:

   a. If you want to input or change the waypoint xy position OR observe the current value:
      (1) Enter: Y
      (2) Press: CONT
      (3) Go to Step 7
   b. If not:
      (1) Enter: N
      (2) Press: CONT
      (3) Go to Step 8

7. When "XY POSITION? E,N" appears in the display area:

   a. If you want to observe the current value:
      (1) Press: CONT
      The waypoint xy position is printed on the hard copy printer.
   b. If you want to change or store new data:
      (1) Enter: X position, Y position in kilometers
      (2) Press: CONT
      The waypoint xy position is printed on the hard copy printer.

8. When "INPUT OR CHANGE LAT/LON?" appears on the display line:

a. If you want to input or change the waypoint latitude and longitude <u>OR</u> observe the current values:
   (1) Enter: Y
   (2) Press: CONT
   (3) Go to Step 9

b. If not:
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 10

9. When "LAT,LON?" appears on the display line:

a. If you want to observe the current value in the table
   (1) Press: CONT
   The waypoint latitude and longitude are printed on the hard copy printer.

b. If you want to input or change the waypoint latitude and longitude
   (1) Enter: Latitude, longitude. Units are degrees, minutes, seconds North latitude and West longitude are positive.
   (2) Press: CONT

10. When "ANOTHER WAYPOINT? Y OR N" appears on the display line:

a. If you want to enter data for another waypoint <u>OR</u> correct data entered:
   (1) Enter: Y
   (2) Press: CONT
   (3) Go to Step 3

b. If not:
   (1) Enter: N
   (2) Press: CONT

11. When "LIST WAYPOINT FILE? Y OR N" appears on the display line:

a. If you want a listing of the waypoint table:
   (1) Enter: Y
   (2) Press: CONT
   The waypoint table is printed on the hard copy printer.

b. If not:
   (1) Enter: N
   (2) Press: CONT

12. When "FILE DATA?" appears on the display line:

a. If you want to store or restore the waypoint table on magnetic tape.
   (1) Insure the tape cartridge used to store the waypoint is in the right hand tape drive
   (2) Enter: Y
   (3) Press: CONT

b. If not:
   (1) Enter: N
   (2) Press: CONT

13. The program function is complete. The CRT is cleared and the program menu is reprinted.

Function: TD MOVE

Special Function Key: K24

Subprograms:

Cart_coord (Xmit(*), Wpt(25,7), Wpt(25,8), Zxmit(*))
Rb (Zp(1), Zp(2), Zxmit(I,1) Zxmit(I,2), Bear(I), Range(I))
Rotate (Glat, Glon, R(*))
Cart (Plat, Plon, X, Y)
FNGlat (Lat, F)
Pseudo (Glat, Glon, R(*))

This program function calculates the change in TD from a waypoint to a position offset from the waypoint. This offset in waypoint position and TD may be applied to the waypoint table. The change in TDs is calculated based on the change in distances to the transmitters.

Function Variables:

C1 - Status variable which indicates if chain data has been read into memory
F1 - Status variable which indicates if the waypoint table has been read into memory
Chain$ - Chain file, e.g. NEUS1, GTLK2, etc.
Ch$ - First four characters of Chain$
Conf - Chain configuration, fifth character of Chain$, 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ
Xmit(*), Power(*), Emis(*) - Transmitter data: geodetic position, power level, emission delay
Wpt(*) - Waypoint table
Wp - Waypoint of interest
Move - Indicates if change in position is to be input as dx, dy or range/bearing
Dx, Dy - Offset in x and y direction from waypoint
R, B - Range and bearing of offset position from waypoint
Zp(*) - Waypoint position and offset position
Zxmit(*) - xy coordinates of transmitters relative to local origin
Bear(*), Range(*) - Ranges and bearings to transmitters
T(*) - Variable to store intermediate calculations and TD offsets
File$ - Waypoint file name
Cor$ - Input variable which indicates if the offset is to be applied as a correction to the waypoint TD in the waypoint table

User Instructions:

Prerequisite function: none. It is assumed that a waypoint file exists.

1. Press K24. The program menu is cleared and "MOVE: CALCULATES $\underline{Td}q = \underline{Td}p+h(\underline{Z}q)-h(\underline{Z}p)$" appears on the CRT.

2. If "CHAIN FILE?" appears on the display line:

   (1) Insure tape containing chain file is in right hand tape drive
   (2) Enter: Chain file (e.g. NEUS2, GTLK1)
   (3) Press: CONT

3. If "WAYPOINT FILE?" appears on the display line:

   a. Insure the tape cartridge containing the waypoint file is in the right hand tape drive.
   b. Enter: Waypoint file name
   c. Press: CONT

4. When "WAYPOINT?" appears on the display line:

   a. Enter: Waypoint of interest
   b. Press: CONT

5. When "Dx/Dy 1 OR RANGE/BEARING 2, 1 OR 2" appears on the display line:

   a. If the offset is to be applied in x and y components:
      (1) Enter: 1
      (2) Press: CONT
      (3) Go to Step 6
   b. If the offset is to be applied as a range and bearing from the waypoint:
      (1) Enter: 2
      (2) Press: CONT
      (3) Go to Step 8

6. When "INPUT DX (KM)" appears on the display line:

   a. Enter: x direction component of the offset in KM
   b. Press: CONT

7. When "INPUT DY (KM)" appears on the display line:

   a. Enter: y direction component of the offset in KM
   b. Press: CONT
   c. Go to Step 9

8. When "RANGE (KM) AND BEARING (DEG)" appears on the display line:

   a. Enter: The range in KM and the bearing in degrees separated by a comma
   b. Press: CONT

9. The effect of offsetting the waypoint on the TDs corresponding to the chain configuration will be printed on the hard copy printer.

10. When "APPLY CORRECTION TO WP? Y OR N" appears on the display line:

   a. If you want to change the waypoint parameters to the offset position:
      (1) Enter: Y
      (2) Press: CONT. (Note: This step does not change the waypoint data stored on magnetic tape. See K23.)
      (3) "WAYPOINT TD AND POSITION CORRECTED" is printed on the hard copy printer.
   b. If not:
      (1) Enter: N
      (2) Press: CONT

11. The program function is complete. The CRT is cleared and the program menu reprinted.

Function: Link Data Files

Special Function Key: K25

Subprograms:

Link (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)
Store (W(*), X(*), Y(*), Z(*), R1(*), R2(*), U$(*), N, F$)

This program function enables multiple data files to be loaded into memory. Range and TD data can be corrected for each file entered. The total number of samples must be equal to or less than 400. The function will automatically ignore any samples which would cause this limit to be exceeded.

Function Variables:

W(*), X(*), Y(*), Z(*) - TD data arrays
R1(*), R2(*) - Miniranger data arrays
U$(*) - Time (Julian day:hours:min:sec) data array
N - Total number of samples
F$ - Final file name

User Instructions:

Prerequisite functions: none

1. Press K25. The program menu is cleared and "LINK DATA FILES" is printed on the CRT.

2. When "FIRST FILE NAME?" appears on the display line:

a. Enter: File name
b. Press: CONT

3. When "CURRENT FILE = (Current file)" and "CORRECT RANGE DATA? Y OR N" appears in the display area:

a. If range data is to be corrected
 (1) Enter: Y
 (2) Press: CONT
b. If not:
 (1) Enter: N
 (2) Press: CONT
 (3) Go to Step 6

4. When "CORRECTION TO R1(METERS)" appears on the display line:

a. Enter: Correction to R1 data in meters
b. Press: CONT

5. When "CORRECTION TO R2(METERS)" appears on the display line:

a. Enter: Correction to R2 data in meters
b. Press: CONT

The current file name and corrections are printed on the hard copy printer. The file name is annotated with an "r".

6. When "CORRECT TD DATA?" appears on the display line:

   a. If TD data is to be corrected:
      (1) Enter: Y
      (2) Press: CONT
   b. If not
      (1) Enter: N
      (2) Press: CONT
      (3) Go to Step 11

7. The start and stop time for the data collected on the current data file are printed on the hard copy printer. When "CORRECTION TO TDW(MICROSEC)?" appears on the display line:

   a. Enter: Correction to TDW in microsec.
   b. Press: CONT

8. When "CORRECTION TO TDX(MICROSEC)?" appears on the display line:

   a. Enter: Correction to TDX in microsec.
   b. Press: CONT

9. When "CORRECTION TO TDY(MICROSEC)?" appears on the display line:

   a. Enter: Correction to TDY in microsec.
   b. Press: CONT

10. When "CORRECTION TO TDZ(MICROSEC)?" appears on the display line:

   a. Enter: Correction to TDZ in microsec.
   b. Press: CONT

The file name is annotated with a "t" and the corrections entered are printed on the hard copy printer.

11. A list of the files linked and total number of samples is printed on the CRT. When "ANOTHER FILE?" appears on the display line:

   a. If another data file is to be entered:
      (1) Enter: Y
      (2) Press: CONT
   b. If not
      (1) Enter: N
      (2) Press: CONT
      (3) Go to Step 13

12. When "FILE NAME?" appears on the display line:

   a. Enter: file name
   b. Press: CONT
   c. Go to Step 3

13. A final list of the files linked and total number of samples is printed on the hard copy printer. When "STORE DATA SET?" appears on the display line:

a. If you want to store the data on a new file
   (1) Enter: Y
   (2) Press: CONT
b. If not
   (1) Enter: N
   (2) Press: CONT
   (3) Go to Step 15

14. When "FILE NAME?" appears on the display line

a. Insert a tape cartridge in the <u>left</u> hand tape drive.
b. Enter: file name
c. Press: CONT

15. If the data is not stored on tape, the file name will be the list of files linked. If the data was stored, the file name returned is the name used to store the data.

16. The program function is complete. The CRT is cleared and the program menu reprinted.

Function: Store Loran-C Chain Data

Special Function Key: K26

Subprograms: None

This program function stores Loran-C chain data (transmitter geodetic positions, transmitter power levels, and secondary emission delays) on a data file for later use. File names are a five character neumonic for the particular Loran-C chain. The first four characters are an abbreviation for the chain (e.g. NEUS, SEUS, GTLK, etc.). The fifth character is a number from 1 to 4 which designates the configuration of three TDs: 1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ.

Function variables:

Chain$ - Five letter chain file
Conf - Chain configuration, fifth character in Chain$

D,M,S - Input variables for degrees, minutes, secondaries when entering latitude or longitude
Power(*) - Transmitter power levels. P(1), P(2), and P(3) are secondary transmitters. P(4) is the master transmitter.
Xmit(*) - Transmitter geodetic positions. Xmit(4,1) and Xmit(4,2) are master latitude and longitude
Emit(*) - Secondary emission delays
L1$, L2$, L3$ - Labels for secondaries, e.g. "TDX"
C$ - Input variable to indicate if data entered is correct

User Instructions:

Prerequisite functions: None

1. Press K26. The program menu is cleared and "STORE CHAIN DATA" is printed on the CRT.

2. "PLACE TAPE CARTRIDGE IN RIGHT HAND TAPE DRIVE" is printed on the CRT. When "INPUT CHAIN FILE NAME" appears on the display line:

a. Enter: File name. The file name is a five character neumonic for the chain of interest. It is important that it be five characters long. The first four characters are an abbreviation for the chain name. Any convenient four letter code is acceptable. The fifth character must be an integer from 1 to 4. The number designates the secondary data to be stored (1 = XYZ, 2 = WXY, 3 = WXZ, 4 = WYZ) in the first three rows of Xmit(*), Power(*) and Emis(*).
b. Press: CONT

Note: If the file name is not a valid name, "INPUT CHAIN FILE NAME" will be repeated.

3. When "INPUT MASTER DATA" and "MASTER LATITUDE? D,M,S" appears in the display area:

   a. Enter: master latitude degrees, minutes, seconds (entries separated by commas). North latitude is assumed.
   b. Press: CONT

4. When "INPUT MASTER LONGITUDE? D,M,S appears in the display area:

   a. Enter: master longitude in degrees, minutes, seconds (separated by commas). West longitude is assumed.
   b. Press: CONT

5. When "POWER LEVEL? KW" appears on the display line:

   a. Enter: power level of master in kilowatts.
   b. Press: CONT

6. The following sequence of input statements will have different secondaries listed depending on the configuration. The statements below are for Conf = 1 (e.g. XYZ). The portion of the statement contained within parentheses are for configurations 2, 3, and 4, respectively.

When "INPUT TDX (TDW,TDW,TDW) DATA" and "LATITUDE? D,M,S" appears in the display area:

   a. Enter: secondary X (W,W,W) latitude in degrees, minutes, seconds
   b. Press: CONT

7. When "LONGITUDE? D,M,S" appears in the display area:

   a. Enter: secondary X (W,W,W) longitude in degrees, minutes, seconds
   b. Press: CONT

8. When "POWER LEVEL? KW" appears in the display area:

   a. Enter: secondary X (W,W,W) power in Kilowatts
   b. Press: CONT

9. When "EMISSION DELAY?" appears on the display line:

   a. Enter: secondary X (W,W,W) emission delay
   b. Press: CONT

10. When "INPUT TDY (TDX,TDX,TDY) DATA" and "LATITUDE? D,M,S" appears in the display area:

   a. Enter: secondary Y (X,X,Y) latitude in degrees, minutes, seconds
   b. Press: CONT

11. When "LONGITUDE? D,M,S" appears on the display line:

a. Enter: secondary Y (X,X,Y) longitude in degrees, minutes, seconds
b. Press: CONT

12. When "POWER LEVEL? KW" appears on the display line:

a. Enter: secondary Y (X,X,Y) power level in kilowatts
b. Press: CONT

13. When "EMISSION DELAY?" appears on the display line:

a. Enter: secondary Y (X,X,Y) emission delay
b. Press: CONT

14. When "INPUT TDZ (TDY,TDZ,TDZ) DATA" and "LATITUDE? D,M,S" appears in the display area:

a. Enter: secondary Z (Y,Z,Z) latitude in degrees, minutes, seconds
b. Press: CONT

15. When "LONGITUDE? D,M,S" appears on the display line:

a. Enter: secondary Z (Y,Z,Z) longitude in degrees, minutes, seconds
b. Press: CONT

16. When "POWER LEVEL? KW" appears on the display line:

a. Enter: secondary Z (Y,Z,Z) power level in kilowatts
b. Press: CONT

17. When "EMISSION DELAY?" appears on the display line:

a. Enter: secondary Z (Y,Z,Z) emission delay
b. Press: CONT

18. A table is printed on the CRT of the data entered. When "IS DATA CORRECT? Y OR N" appears on the display line:

a. If the data is correct:
    (1) Ensure a tape cartridge is in the right hand tape drive
    (2) Enter: Y
    (3) Press: CONT
b. If not:
    (1) Enter: N
    (2) Press: CONT
    (3) Go to Step 3

19. The program function is complete. The CRT is cleared and the program menu is printed on the CRT.

Function: Reflect TDs to Waypoint

Special Function Key: K27

Subprograms:

Cart coord(Xmit(*), Wpt(25,7) Wpt(25,8) Zxmit(*))
Reflect (W, Wpt(*), W(*), X(*), Y(*), Z(*), Zx(*), Zy(*), Conf, Zxmit(*), N, V)
Rotate (Glat, Glon, R(*))
Cart (Plat, Plon, X, Y)
FNGlat (Lat, F)
Pseudo (Glat, Glon, R(*), Plat, Plon)
Rb (x1, y1, x2, y2, B, R)

This program function reflects each TD sample to the waypoint selected based on the position of the measurement relative to the waypoint. The resultant data arrays are averaged to produce an estimate of the waypoint TDs.

Function variables:

Cl - status variable which indicates if chain data is in memory
Fl- status variable which indicates if the waypoint table is in memory
Chain$ - chain file name
Ch$ - first four characters if Chain$
Conf - chain configuration, fifth character in Chain$
Xmit(*), Power(*), Emis(*) - transmitting station parameters: geodetic position, power level and emission delay
Wpt(*) - waypoint table
File$ - waypoint file name
Zxmit(*) - transmitter xy coordinates referenced to local origin
W - waypoint of interest
W(*), X(*), Y(*), Z(*) - TD data arrays
Zx(*), Zy(*) - measurement position coordinates calculated from Miniranger data
N - Number of samples
V - Velocity of propagation
F$ - data file name

<u>User Instructions</u>:

Prerequisite functions:

K0 - Read Data
K16 - Edit Data (Optional)
K1 - Calculate XY Position from Range Data

1. Press K27. The CRT is cleared and "REFLECT TDs TO WAYPOINT" is printed on the CRT.

2. If "CHAIN FILE?" appears on the display line:

(1) Enter: Chain file name
(2) Press: CONT

3. If "WAYPOINT FILE NAME?" appears on the display line:

   a. Enter: waypoint file name
   b. Press: CONT

4. When "WAYPOINT WHERE DATA IS TO BE REFLECTED?" appears on the display line:

   a. Enter: waypoint number
   b. Press: CONT

5. The data file name is annotated with a right bracket symbol and "RESULTS OF REFLECTING FILE (F$) TDS TO WAYPOINT (W)" is printed on the hard copy printer followed by the statistics table generated by function K2.

6. The program function is complete. The CRT is cleared and the program menu is reprinted on the CRT.

Note: The data contained in the TD arrays (W(*), X(*), Y(*), Z(*))) has been changed.

Function: Create or Read Reference Station File

Special Function Key: K28

Subprograms:

Ref-file (Ref(*), Ref$(*), F2, Rp$)

This program function is used to store reference station coordinates on a file for use in converting range data to xy coordinates.

Function variables:

Ref(*) - reference station table. Rows 1 and 2 contain the state plane or local coordinates of the two reference stations. Row 3 contains the coordinates of the local origin.
Ref$(*) - reference station identifiers
Rp$ - reference station file name

User Instructions:

Prerequisite functions: None

1. Press K28. The CRT is cleared and "CREATE OR READ REFERENCE STATION FILE" is printed on the CRT.

2. When "REFERENCE POSITION FILE NAME?" appears on the display line:

   a. Enter: file name
   b. Press: CONT

3. When "DOES THIS FILE CURRENTLY EXIST ON TAPE?" appears on the display line:

   a. If the file has been previously created:
      (1) Ensure the tape cartridge is in the right hand drive
      (2) Enter: Y
      (3) Press: CONT
   b. If not:
      (1) Ensure the tape cartridge is in the right hand drive
      (2) Enter: N
      (3) Press: CONT

4. When "NEW OR ADDITIONAL DATA? Y OR N" appears on the display line:

   a. If data is to be entered:
      (1) Enter: Y
      (2) Press: CONT
   b. If not:
      (1) Enter: N
      (2) Press: CONT
      (3) Go to Step 9

5. When "REFERENCE NO? 1-3" appears on the display line:

   a. Enter: reference station number. Numbers 1 and 2 are for the two reference stations. Reference number 3 is the local origin.
   b. Press: CONT

6. When "LABEL?" appears on the display line:

   a. Enter: reference station identifier
   b. Press: CONT

7. When "X-POSITION?" appears on the display line:

   a. Enter: the X(East) position of the reference station in KM with respect to the local origin or the state plane coordinates in KM.
   b. Press: CONT

8. When "Y-POSITION?" appears on the display line:

   a. Enter: the Y(North) position of the reference station in KM with respect to the local origin or the state plane coordinates in KM.
   b. PressP CONT
   c. Go to Step 4

9. When "LIST DATA?" appears on the display line:

   a. If you want a list of the reference station data:
      (1) Enter: Y
      (2) Press: CONT. The reference station data is printed on the hard copy printer.
   b. If not:
      (1) Enter: N
      (2) Press: CONT

10. When "FILE DATA?" appears on the display line:

   a. If you want to file the data:
      (1) Enter: Y
      (2) Press: CONT
   b. If not:
      (1) Enter: N
      (2) Press: CONT

11. The program function is complete. The CRT is cleared and the program menu is reprinted.

Program COMPAR Subroutines

Name	Starting Line NO	Program Functions
Bear	10710	K20
Cart	21750	K17,21,22,24,27
Cart_coord	22750	K17,21,22,24,27
Compar	26180	K3
Ct_at	13970	K1,17
Delete	14990	K16
Delete_blok	17560	K16
Delete_rg	22380	K16
Delete_td	17980	K16
Dif	18900	K3
Fehgt	12550	K17,22
FNGlat	21900	K17,21,22,24,27
G_mat	11150	K17,20,22
Hi_lo	9360	K2,3,5,18,19,21
Link	15750	K25
Patct	14240	K19
Plot	9440	K4
Plot_err	19130	K3
Position	18830	K1
Print	9040	K1,K2
Pseudo	21410	K17,21,22,24,27
Range	10860	K20
Rb	25510	K17,22,24
Read	15400	K0
Ref_file	21960	K28
Reflect	26470	K27
Reg	8710	K1,2,17
Rotate	21600	K17,21,22,24,27
Rplot	9760	K5
Stat_mat	16570	K1,2,3,17,21
Store	17130	K16,25
Td	11670	K20
Td2	24650	K17
Td3	23440	K17
T_errplot	20500	K21
Track	18400	K17
Triangle	18600	K1
Warp	19800	K21
Wpfile	12030	K23
Wpf2	25110	K17,22
Wp2	23770	K17,22
Wp3	23030	K17,22
X_errplot	19500	K18
Xplot	13130	K18

```
1     !     COMPARE PROGRAM
2           !     10 APRIL 1982
3           !
10   OPTION BASE 1
20    DEG
30   DIM W(400),X(400),Y(400),Z(400),R1(400),R2(400),U$(400)[14]
40   DIM P(4),O_set(4),Stat(4,4),Cov(3,6),S(4,6),R(3,6),O(4)
50   DIM S1(4,6),S2(4,6),Wp(4,6),Wpt(25,8),T(3),Zd(2),F$[80]
60   DIM Xmit(4,2),Emis(3),Bear(4),Range(4),Td(3),Power(4),G123(2,3),G12(2,2
)
70   DIM G13(2,2),G23(2,2),Zxmit(4,2),Zp(2),Aa(3,2),Pos_x(400),Pos_y(400),Tp
(3,1),Tq(3,1)
80   DIM Zq(2,1),Zpp(2,1),At(400),Ct(400),Tpp(2,1),Tqq(2,1),Zqq(2,1)
90   DIM Zx(400),Zy(400),Zb(2),Rotate(3,3),Gx(4,2),Px(4,2),Axy(2,2)
100 DIM Tw(2),Zm(2),Zt(2),Ref$(3),Ref(3,2),Att(400),Ctt(400)
110 DIM Zxx(2,1),Zxm(3,2),G(2,2)
120   V=.299792458/1.000338
130   F=.00335278
230   !
250 Keys:  !
260   ON KEY #0 GOTO K0
270   ON KEY #1 GOTO K1
280   ON KEY #2 GOTO K2
290   ON KEY #3 GOTO K3
300   ON KEY #4 GOTO K4
310   ON KEY #5 GOTO K5
320   ON KEY #16 GOTO K16
330   ON KEY #17 GOTO K17
340   ON KEY #18 GOTO K18
350   ON KEY #19 GOTO K19
360   ON KEY #20 GOTO K20
370   ON KEY #21 GOTO K21
380   ON KEY #22 GOTO K22
390   ON KEY #23 GOTO K23
400   ON KEY #24 GOTO K24
410   ON KEY #25 GOTO K25
420   ON KEY #26 GOTO K26
430   ON KEY #27 GOTO K27
440   ON KEY #28 GOTO K28
450 Menu:  !
460   MASS STORAGE IS ":T15"
470   PRINTER IS 16
480   PRINT PAGE
490   PRINT "      AUGMENTED SURVEY: ANALYSIS 2   "
500   PRINT "K0:READ TRACKLINE FILE DATA"
510   PRINT "K1:CONVERT RANGE DATA TO XY POSITIONS"
520   PRINT "K2:STATS AND REGRESSION OF TD DATA"
530   PRINT "K3:COMPARE MINI-RANGER AND LORAN-C XY POSITIONS"
540   PRINT "K4:PLOT TD DATA WITH REGRESSION LINE"
550   PRINT "K5:PLOT RESIDUALS OF TD DATA"
560   PRINT "K16:EDIT DATA"
570   PRINT "K17:CONVERT TDs TO XY AND AT/CT"
580   PRINT "K18:PLOT XY DATA"
590   PRINT "K19:PLOT AT/CT DATA"
600   PRINT "K20:PREDICT TD"
```

```
610    PRINT "K21:CALCULATE TD GRID WARP"
620    PRINT "K22:DAISY CHAIN WAYPOINTS"
630    PRINT "K23:FILE OR READ WP DATA"
640    PRINT "K24:TD MOVE"
650    PRINT "K25:LINK DATA FILES"
660    PRINT "K26:STORE CHAIN DATA"
670    PRINT "K27:REFLECT TDS TO WAYPOINT"
680    PRINT "K28:FILE OR READ REFERENCE STATION DATA"
690 Loop: GOTO Loop
```

```
710 Ko: !    ^      READ DATA FILE

720      PRINT PAGE
730      PRINT " READ DATA FILE "
740      MASS STORAGE IS ":T14"
750      PRINTER IS 0
760      FIXED 2
770      CALL Read(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
780      PRINT "FILE=";F$;TAB(30);"SAMPLES=";N
790      Range$="N"
800      INPUT "CORRECT RANGE DATA?,Y OR N",Range$
810      IF Range$="N" THEN Td_cor
820      INPUT "CORRECTION TO R1(METERS)?",R1c
830      INPUT "CORRECTION TO R2(METERS)?",R2c
840      PRINT "R1 CORRECTION=";R1c;TAB(30);"R2 CORRECTION=";R2c
850      FOR I=1 TO N
860      R1(I)=R1(I)+R1c
870      R2(I)=R2(I)+R2c
880      NEXT I
890      F$=F$&"r"
900 Td_cor:    !
910      Td_cor$="N"
920      INPUT "CORRECT TDs? Y/N",Td_cor$
930      IF Td_cor$<>"Y" THEN Menu
940      PRINT "START TIME=";U$(1);TAB(30);"STOP TIME=";U$(N)
950      INPUT "CORRECTION TO TDW(MICROSEC)?",Wcor
960      INPUT "CORRECTION TO TDX(MICROSEC)?",Xcor
970      INPUT "CORRECTION TO TDY(MICROSEC)?",Ycor
980      INPUT "CORRECTION TO TDZ(MICROSEC)?",Zcor
990      FOR I=1 TO N
1000     W(I)=W(I)+Wcor
1010     X(I)=X(I)+Xcor
1020     Y(I)=Y(I)+Ycor
1030     Z(I)=Z(I)+Zcor
1040     NEXT I
1050     PRINT "TD CORRECTIONS:";TAB(20);"Wcor=";Wcor;TAB(35);"Xcor=";Xcor;TA
B(50);"Ycor=";Ycor;TAB(65);"Zcor=";Zcor
1060     F$=F$&"t"
1070     GOTO Menu
1080     !
```

```
1100 K1:! ^     CONVERT RANGES TO XY
1110  PRINT PAGE
1120  MASS STORAGE IS ":T15"
1130  FIXED 4
1140  PRINT " CONVERT RANGES TO XY POSITION "
1150  PRINTER IS 0
1160  INPUT "POSITION REFERENCE FILE NAME?",Rp$
1170  PRINT "POSITION REFERENCE FILE NAME=";Rp$
1180  PRINTER IS 16
1190  ASSIGN #1 TO Rp$
1200  READ #1;Ref$(*),Ref(*)
1210  PRINT "REFERENCE POSITIONS:"
1220  FOR I=1 TO 2
1230  PRINT I,Ref$(I),Ref(I,1),Ref(I,2)
1240  NEXT I
1250  PRINT "LOCAL GRID ORIGION:"
1260  PRINT "3",Ref$(3),Ref(3,1),Ref(3,2)
1270  Zb(1)=Ref(1,1)-Ref(3,1)
1280  Zb(2)=Ref(1,2)-Ref(3,2)
1290  Zc(1)=Ref(2,1)-Ref(3,1)
1300  Zc(2)=Ref(2,2)-Ref(3,2)
1310  ! CALCULATE RANGE AND BEARING BETWEEN REF POINTS
1320  R=SQR((Zc(1)-Zb(1))^2+(Zc(2)-Zb(2))^2)
1330  Alpha=ATN((Zc(1)-Zb(1))/(Zc(2)-Zb(2)))
1340  IF Zc(2)-Zb(2)<0 THEN Alpha=Alpha+180
1350  IF Alpha<0 THEN Alpha=Alpha+360
1360  PRINT "RANGE=";R,"BEARING=";Alpha
1370  Point=1
1380  INPUT "IS R1 AT POINT B1 OR C2? 1 OR 2",Point
1390  Point$="B"
1400  IF Point=2 THEN Point$="C"
1410  PRINT "R1 TRANSPONDER IS AT POINT ";Point$
1420  MAT Zx=(0)
1430  MAT Zy=(0)
1440  Sign=1
1441  INPUT "ENTER SIGN, +1 OR -1",Sign
1442  Q=0
1443  INPUT "DATA SAMPLE TO CHANGE SIGN",Q
1450  PRINT "CONVERTING DATA"
1460  FOR I=1 TO N
1461  IF I=Q THEN Sign=-Sign
1470  IF Point=1 THEN Rc=R1(I)
1480  IF Point=1 THEN Rb=R2(I)
1490  IF Point=2 THEN Rc=R2(I)
1500  IF Point=2 THEN Rb=R1(I)
1510  CALL Triangle(R,Rb/1000,Rc/1000,A,B,C)
1520  DISP I,A
1530  CALL Position(Zb(*),Alpha,Rc/1000,Sign,B,Zx(I),Zy(I))
1540  NEXT I
1550  PRINTER IS 0
1560   CALL Stat_mat(R1(*),R2(*),Zx(*),Zy(*),Stat(*),Cov(*),O_set(*),N)
1570   CALL Reg(Stat(*),Cov(*),S(*),R(*),O_set(*))
1580   CALL Print(Stat(*),O_set(*),Cov(*),R(*),N,2)
1590  IF F1 THEN 1640
```

```
1600  INPUT "WAYPOINT FILE NAME?",File$
1610  ASSIGN #1 TO File$
1620  READ #1;Wpt(*)
1630  Fi=1
1640  INPUT "WAYPOINT TO?",Wt
1650  INPUT "WAYPOINT FROM?",Wf
1660   CALL Ct_at(Zx(*),Zy(*),Ctt(*),Att(*),Wpt(Wt,5),Wpt(Wt,6),Wpt(Wf,5),W
pt(Wf,6),N,R,Angle)
1670  GOTO Menu
1690  !
```

```
1700 K2:! ^        STATS AND REGRESSION
1710   MASS STORAGE IS ":T14"
1720   PRINT PAGE
1730   PRINT " STATISTICS AND REGRESSION OF TD DATA "
1740   PRINTER IS 0
1750   CALL Hi_lo(W(*),N,Bw,Lw)
1760   CALL Hi_lo(X(*),N,Bx,Lx)
1770   CALL Hi_lo(Y(*),N,By,Ly)
1780   CALL Hi_lo(Z(*),N,Bz,Lz)
1790   CALL Stat_mat(W(*),X(*),Y(*),Z(*),Stat(*),Cov(*),O_set(*),N)
1800   CALL Reg(Stat(*),Cov(*),S(*),R(*),O_set(*))
1810   PRINT "TRACKLINE=";F$;TAB(20);" START TIME=";U$(1);TAB(50);" STOP TI
ME=";U$(N)
1820   CALL Print(Stat(*),O_set(*),Cov(*),R(*),N,1)
1830   GOTO Menu
1840  !
```

```
1850 K3:! ^          COMPARE MINI-RANGER AND
1860   PRINT PAGE
1870   PRINT " COMPARE MINI-RANGER AND LORAN-C POSITION DATA "
1880  CALL Compar(Pos_x(*),Pos_y(*),Zx(*),Zy(*),At(*),Ct(*),Att(*),Ctt(*),F
$,N,R)
1890  GOTO Menu
```

```
1900 K4: !  ^      PLOT TD DATA WITH REGRESS
1910     PRINT PAGE
1920     PRINT " PLOT TD DATA WITH REGRESSION LINES "
1930     INPUT "PLOT? WX1,WY2,WZ3,XY4,XZ5,YZ6",Plot
1940     ON Plot GOTO Wx,Wy,Wz,Xy,Xz,Yz
1950 Wx:   PRINT LIN(3)
1960     CALL Plot(Lw,Bw,Lx,Bx,S(1,1),S(2,1),W(*),X(*),R(1,1),N,"TDX","TDW",
F$)
1970     PRINT LIN(3)
1980     GOTO Menu
1990 Wy:    PRINT LIN(3)
2000     CALL Plot(Lw,Bw,Ly,By,S(1,2),S(2,2),W(*),Y(*),R(1,2),N,"TDY","TDW",
F$)
2010     PRINT LIN(3)
2020     GOTO Menu
2030 Wz:    PRINT LIN(3)
2040     CALL Plot(Lw,Bw,Lz,Bz,S(1,3),S(2,3),W(*),Z(*),R(1,3),N,"TDZ","TDW",
F$)
2050     PRINT LIN(3)
2060     GOTO Menu
2070 Xy:   PRINT LIN(3)
2080     CALL Plot(Lx,Bx,Ly,By,S(1,4),S(2,4),X(*),Y(*),R(1,4),N,"TDY","TDX",
F$)
2090     PRINT LIN(3)
2100     GOTO Menu
2110 Xz:   PRINT LIN(3)
2120     CALL Plot(Lx,Bx,Lz,Bz,S(1,5),S(2,5),X(*),Z(*),R(1,5),N,"TDZ","TDX",
F$)
2130     PRINT LIN(3)
2140     GOTO Menu
2150 Yz:   PRINT LIN(3)
2160     CALL Plot(Ly,By,Lz,Bz,S(1,6),S(2,6),Y(*),Z(*),R(1,6),N,"TDZ","TDY",
F$)
2170     GOTO Menu
2180   !
```

```
2210 KS:1 ^      PLOT RESIDUALS
2220     PRINT PAGE
2230     Hc=0
2240     PRINT " PLOT RESIDUALS "
2250     INPUT "PLOT? WX1,WY2,WZ3,XY4,XZ5,YZ6",Plot
2260     ON Plot GOTO Rwx,Rwy,Rwz,Rxy,Rxz,Ryz
2270 Rwx:  IF R(3,1)=2 THEN Skip1
2280     CALL Rplot(W(*),X(*),S(*),R(*),1,N,"TDW","TDX",F$)
2290     PRINT LIN(3)
2300     GOTO Menu
2310 Skip1:CALL Rplot(X(*),W(*),S(*),R(*),1,N,"TDW","TDX",F$)
2320     PRINT LIN(3)
2330     GOTO Menu
2340 Rwy:  IF R(3,2)=2 THEN Skip2
2350     CALL Rplot(W(*),Y(*),S(*),R(*),2,N,"TDW","TDY",F$)
2360     PRINT LIN(3)
2370     GOTO Menu
2380 Skip2:CALL Rplot(Y(*),W(*),S(*),R(*),2,N,"TDW","TDY",F$)
2390     PRINT LIN(3)
2400     GOTO Menu
2410 Rwz:  IF R(3,3)=2 THEN Skip3
2420     CALL Rplot(W(*),Z(*),S(*),R(*),3,N,"TDW","TDZ",F$)
2430     PRINT LIN(3)
2440     GOTO Menu
2450 Skip3:CALL Rplot(Z(*),W(*),S(*),R(*),3,N,"TDW","TDZ",F$)
2460     PRINT LIN(3)
2470     GOTO Menu
2480 Rxy:  IF R(3,4)=2 THEN Skip4
2490     CALL Rplot(X(*),Y(*),S(*),R(*),4,N,"TDX","TDY",F$)
2500     PRINT LIN(3)
2510     GOTO Menu
2520 Skip4:CALL Rplot(Y(*),X(*),S(*),R(*),4,N,"TDX","TDY",F$)
2530     PRINT LIN(3)
2540     GOTO Menu
2550 Rxz:  IF R(3,5)=2 THEN Skip5
2560     CALL Rplot(X(*),Z(*),S(*),R(*),5,N,"TDX","TDZ",F$)
2570     PRINT LIN(3)
2580     GOTO Menu
2590 Skip5:CALL Rplot(Z(*),X(*),S(*),R(*),5,N,"TDX","TDZ",F$)
2600     PRINT LIN(3)
2610     GOTO Menu
2620 Ryz:  IF R(3,6)=2 THEN Skip6
2630     CALL Rplot(Y(*),Z(*),S(*),R(*),6,N,"TDY","TDZ",F$)
2640     PRINT LIN(3)
2650     GOTO Menu
2660 Skip6:CALL Rplot(Z(*),Y(*),S(*),R(*),6,N,"TDY","TDZ",F$)
2670     PRINT LIN(3)
2680     GOTO Menu
2690     !
```

```
2720 K16: !  ^     EDIT TD DATA
2730      MASS STORAGE IS ":T14"
2731      PRINTER IS 16
2740     PRINT PAGE
2750     PRINT " EDIT TD DATA "
2751     Edit=5
2752     S$="N"
2760     INPUT "SINGLE LINE1,BLOCK2,TD-CLIP3,OR RANGE4?",Edit
2770     ON Edit GOTO Single,Blok,Clip,Range,More
2771     GOTO More
2780 Single:CALL Delete(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
2790     GOTO More
2800 Blok:CALL Delete_blok(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
2810     GOTO More
2820 Clip:CALL Delete_td(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
2830     GOTO More
2840 Range:CALL Delete_rg(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
2850 More: !
2851     INPUT "EDIT MORE DATA? Y OR N",S$
2852     IF UPC$(S$[1;1])="Y" THEN K16
2853     INPUT "LIST DATA? Y OR N",S$
2854     IF UPC$(S$[1;1])<>"Y" THEN Stor
2855     FIXED 2
2856     FOR I=1 TO N
2857     PRINT I;W(I);X(I);Y(I);Z(I);R1(I);R2(I);U$(I)
2858     NEXT I
2860 Stor:F$=F$&"e"
2868     INPUT " STORE EDITED DATA? Y OR N ",S$
2870     IF UPC$(S$[1;1])<>"Y" THEN Menu
2880     CALL Store(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
2890     GOTO Menu
2920     !
```

```
2930 K17:! ^   CONVERT TD TO XY
2940      PRINT PAGE
2950      PRINT " CONVERT TD TO XY "
2960      MASS STORAGE IS ":T15"
2970      MAT Pos_x=(0)
2980      MAT Pos_y=(0)
2990      MAT Ct=(0)
3000      MAT At=(0)
3010 Chain:     !
3020      IF Ci=1 THEN Wp_file
3040      INPUT "CHAIN FILE",Chain$
3060      IF LEN(Chain$)<>5 THEN Chain
3070      ASSIGN #1 TO Chain$
3075      Ci=1
3080      READ #1;Xmit(*),Power(*),Emis(*)
3090 Wp_file:     !
3100      IF Fi=1 THEN Xmit_coord
3110      INPUT "WAYPOINT FILE NAME?",File$
3120      Fi=1
3130      ASSIGN #1 TO File$
3140      READ #1;Wpt(*)
3150 Xmit_coord:     !
3160      CALL Cart_coord(Xmit(*),Wpt(25,7),Wpt(25,8),Zxmit(*))
3170 Waypoint:     !
3180      INPUT "WAYPOINT TO ?",Wt
3190      INPUT "WAYPOINT FROM ?",Wf
3200 Range_bearing:     !
3210      FOR I=1 TO 4
3220      CALL Rb(Wpt(Wt,5),Wpt(Wt,6),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I)
)
3230      NEXT I
3240 G_mat:     !
3250      Zp(1)=Wpt(Wt,5)
3260      Zp(2)=Wpt(Wt,6)
3270      CALL G_mat(Power(*),V,Range(*),Bear(*),Zp(*),Aa(*),G123(*),G12(*),
G23(*),G13(*))
3280 Configuration:     !
3290      PRINTER IS 0
3310      Conf=VAL(Chain$[5,5])
3320      Ch$=Chain$[1,4]
3340      INPUT "TWO OR THREE TD SOLUTION?",So
3350      IF So=2 THEN So2
3360      IF So=3 THEN So3
3370 So3:     ! THREE TD SOLUTION
3380      CALL Wp3(Wpt(*),Conf,Wt,Tp(*),Zpp(*),L$)!WAYPOINT TD
3390      PRINT "THREE TD SOLUTION, CHAIN=";Ch$;"   LOPs=";L$
3400      FOR I=1 TO N
3410      CALL Td3(W(*),X(*),Y(*),Z(*),I,Conf,Tq(*))
3420      CALL Fehg(Zxmit(*),Zpp(*),Tp(*),Tq(*),G123(*),Zq(*),V,3)
3430      Pos_x(I)=Zq(1,1)
3440      Pos_y(I)=Zq(2,1)
3450      DISP I
3460      NEXT I
3470      ! CALCULATE POSITION OF FROM WAYPOINT (Wf)
```

```
3480     CALL Wp3(Wpt(*),Conf,Wf,Tq(*),Zxx(*),L$)!WAYPOINT FROM TD
3490     CALL Fehg(Zxmit(*),Zpp(*),Tp(*),Tq(*),G123(*),Zq(*),V,3)
3500     GOTO Ct_at
3510 So2:    ! TWO TD SOLUTION
3520     CALL Wp2(Wpt(*),Conf,Wt,Tpp(*),Pair,Zpp(*),G12(*),G23(*),G13(*),G(
*),L$,Zxmit(*),Zxm(*))
3530     PRINT "TWO TD SOLUTION, CHAIN=";Ch$;"   LOPs=";L$
3540     FOR I=1 TO N
3550     CALL Td2(W(*),X(*),Y(*),Z(*),I,Conf,Pair,Tqq(*))
3560     CALL Fehg(Zxm(*),Zpp(*),Tpp(*),Tqq(*),G(*),Zq(*),V,2)
3570     Pos_x(I)=Zq(1,1)
3580     Pos_y(I)=Zq(2,1)
3590     DISP I
3600     NEXT I
3610     ! CALCULATE POSITION OF FROM WAYPOINT(Wf)
3620     CALL Wpf2(Wpt(*),Conf,Pair,Wf,Tqq(*))
3630     CALL Fehg(Zxm(*),Zpp(*),Tpp(*),Tqq(*),G(*),Zq(*),V,2)
3640     ! CALCULATE CROSSTRACK AND ALONGTRACK POSITIONS
3650 Ct_at:  CALL Ct_at(Pos_x(*),Pos_y(*),Ct(*),At(*),Wpt(Wt,5),Wpt(Wt,6),Z
q(1,1),Zq(2,1),N,R,Angle)
3660     DISP R,Angle
3670     CALL Stat_mat(Pos_x(*),Pos_y(*),Ct(*),At(*),Stat(*),Cov(*),O_set(*
),N)
3680     CALL Reg(Stat(*),Cov(*),S(*),R(*),O_set(*))
3690     PRINTER IS 0
3700  CALL Track(Stat(*),O_set(*),Cov(*),R(*),Sample,F$,Wt,Wf,Angle,N)
3710     GOTO Menu
3720     !
```

```
3740 K18:! ^     PLOT XY DAT:

3750     PRINT PAGE
3760     PRINT " PLOT XY DATA "
3770     INPUT "MINI-RANGER1 ,CONVERTED LORAN-C2,BOTH3,OR ERROR PLOT
4?",P1
3780     ON P1 GOTO 3830,3790,3870,3920
3790     CALL Hi_lo(Pos_x(*),N,Maxx,Minx)
3800     CALL Hi_lo(Pos_y(*),N,Maxy,Miny)
3810     CALL Xplot(1,Pos_x(*),Pos_y(*),Minx,Maxx,Miny,Maxy,N,1,F$,Ch$,L$,P
1)
3820     GOTO Menu
3830     CALL Hi_lo(Zx(*),N,Maxx,Minx)
3840     CALL Hi_lo(Zy(*),N,Maxy,Miny)
3850     CALL Xplot(1,Zx(*),Zy(*),Minx,Maxx,Miny,Maxy,N,1,F$,Ch$,L$,P1)
3860     GOTO Menu
3870     CALL Hi_lo(Pos_x(*),N,Maxx,Minx)
3880     CALL Hi_lo(Pos_y(*),N,Maxy,Miny)
3890     CALL Xplot(2,Pos_x(*),Pos_y(*),Minx,Maxx,Miny,Maxy,N,1,F$,Ch$,L$,P
1)
3900     CALL Xplot(1,Zx(*),Zy(*),Minx,Maxx,Miny,Maxy,N,2,F$,Ch$,L$,P1)
3910     GOTO Menu
3920     CALL Hi_lo(Pos_x(*),N,Maxx,Minx)
3930     CALL Hi_lo(Pos_y(*),N,Maxy,Miny)
3940     CALL X_errplot(1,Zx(*),Zy(*),Pos_x(*),Pos_y(*),Minx,Maxx,Miny,Maxy
,N)
3950     GOTO Menu
3960  !
```

```
3970 K19:1  ^    PLOT CT/AT DATA

3980      PRINT PAGE
3990      PRINT " PLOT ALONG TRACK CROSS TRACK DATA "
4000      INPUT "MINIRANGER1, LORAN-C2 OR BOTH3? 1,2 OR 3",Data
4010      ON Data GOTO 4020,4060,4100
4020      CALL Hi_lo(Ctt(*),N,Maxc,Minc)
4030      CALL Hi_lo(Att(*),N,Maxa,Mina)
4040      CALL Patct(Att(*),Ctt(*),N,Mina,Maxa,Minc,Maxc,Wt,Wf,F$,R,1,Data)
4050      GOTO Menu
4060      CALL Hi_lo(Ct(*),N,Maxc,Minc)
4070      CALL Hi_lo(At(*),N,Maxa,Mina)
4080      CALL Patct(At(*),Ct(*),N,Mina,Maxa,Minc,Maxc,Wt,Wf,F$,R,1,Data)
4081      INPUT "LIST OF ALONG/CROSS TRACK DATA? Y OR N",List$
4082      IF UPC$(List$[1;1])<>"Y" THEN Menu
4083      PRINTER IS 0
4084      PRINT "FILE= ";F$
4085      PRINT LIN(1);"SAMPLE";TAB(20);"ALONG TRACK (KM)";TAB(40);"CROSS TR
ACK (M)"
4086      PRINT LIN(1)
4087      FOR I=1 TO N
4088      PRINT I;TAB(25);At(I);TAB(45);Ct(I)*1000
4089      NEXT I
4090      GOTO Menu
4100      CALL Hi_lo(Ct(*),N,Maxc,Minc)
4110      CALL Hi_lo(Ctt(*),N,Maxcc,Mincc)
4120      Maxc=MAX(Maxc,Maxcc)
4130      Minc=MIN(Minc,Mincc)
4140      CALL Hi_lo(At(*),N,Maxa,Mina)
4150      CALL Hi_lo(Att(*),N,Maxaa,Minaa)
4160      Maxa=MAX(Maxa,Maxaa)
4170      Mina=MIN(Mina,Minaa)
4180      CALL Patct(Att(*),Ctt(*),N,Mina,Maxa,Minc,Maxc,Wt,Wf,F$,R,1,Data)
4190      CALL Patct(At(*),Ct(*),N,Mina,Maxa,Minc,Maxc,Wt,Wf,F$,R,2,Data)
4200      GOTO Menu
4230      !
```

```
4240 K20:1 ^      PREDICT TD
4250      PRINT PAGE
4260      PRINT " TD PREDICTION "
4270      MASS STORAGE IS ":T15"
4280      PRINTER IS 0
4290      FIXED 3
4300      IF C1=1 THEN Wpfile
4320 Getch:INPUT "CHAIN FILE",Chain$
4340      IF LEN(Chain$)<>5 THEN Getch
4350      ASSIGN #1 TO Chain$
4355      C1=1
4360      READ #1;Xmit(*),Power(*),Emis(*)
4370 Wpfile: INPUT "IS POINT OF INTEREST ON WAYPOINT FILE? Y OR N",P$
4380      IF P$="N" THEN Input
4390      IF F1=1 THEN Jump
4400      INPUT "WP FILE NAME?",File$
4410      F1=1
4420      ASSIGN #1 TO File$
4430      READ #1;Wpt(*)
4440 Jump:     INPUT "WAYPOINT NO.?",Wn
4450      Lat=Wpt(Wn,7)
4460      Lon=Wpt(Wn,8)
4470      GOTO Td
4480 Input:    INPUT " INPUT LAT IN DEGREES,MIN,SEC",D,M,S
4490      Lat=D+M/60+S/3600
4500      INPUT "INPUT LON IN DEGREES,MIN,SEC",D,M,S
4510      Lon=-(D+M/60+S/3600)
4520 Td:    FOR I=1 TO 4
4530      CALL Range(Lat,Lon,Xmit(I,1),Xmit(I,2),Range(I))
4540      CALL Bear(Lat,Lon,Xmit(I,1),Xmit(I,2),Bear(I))
4550      NEXT I
4560      CALL Td(Range(*),V,Emis(*),Td(*))
4570      CALL G_mat(Power(*),V,Range(*),Bear(*),Zp(*),Aa(*),G123(*),G12(*),
G23(*),G13(*))
4580 Gdop:    !
4590      Gdop12=(G12(1,1)^2+G12(1,2)^2+G12(2,1)^2+G12(2,2)^2)^.5
4600      Gdop13=(G13(1,1)^2+G13(1,2)^2+G13(2,1)^2+G13(2,2)^2)^.5
4610      Gdop23=(G23(1,1)^2+G23(1,2)^2+G23(2,1)^2+G23(2,2)^2)^.5
4620      Gdop123=(G123(1,1)^2+G123(1,2)^2+G123(1,3)^2+G123(2,1)^2+G123(2,2)
^2+G123(2,3)^2)^.5
4630      FIXED 4
4650      Ch$=Chain$[1,4]
4660      PRINT "CHAIN=";Ch$
4670      PRINT "LAT=";Lat;TAB(15);"LON=";Lon
4680      FIXED 3
4700      Conf=VAL(Chain$[5,5])
4710      ON Conf GOTO Txyz,Twxy,Twxz,Twyz
4720 Txyz:PRINT "PREDICTED: TDX=";Td(1);TAB(35);"TDY=";Td(2);TAB(55);"TDZ="
;Td(3)
4730      IF P$="N" THEN 4750
4740      PRINT "SURVEYED:  TDX=";Wpt(Wn,2);TAB(35);"TDY=";Wpt(Wn,3);TAB(55)
;"TDZ=";Wpt(Wn,4)
4750      PRINT TAB(20);"RANGE(KM)";TAB(40);"BEARING(DEG)"
4760      PRINT "MASTER";TAB(20);Range(4);TAB(40);Bear(4)
```

```
4770      PRINT "X-RAY";TAB(20);Range(1);TAB(40);Bear(1)
4780      PRINT "YANKEE";TAB(20);Range(2);TAB(40);Bear(2)
4790      PRINT "ZULU";TAB(20);Range(3);TAB(40);Bear(3)
4800     PRINT TAB(22);"XYZ";TAB(37);"XY";TAB(52);"XZ";TAB(67);"YZ"
4810     PRINT "GDOP(M/NANOSEC)";TAB(20);Gdop123;TAB(35);Gdop12;TAB(50);Gdop
13;TAB(65);Gdop23
4820      GOTO Menu
4830 Twxy:PRINT "PREDICTED: TDW=";Td(1);TAB(35);"TDX=";Td(2);TAB(55);"TDY="
;Td(3)
4840      IF P$="N" THEN 4860
4850      PRINT "SURVEYED:  TDW=";Wpt(Wn,1);TAB(35);"TDX=";Wpt(Wn,2);TAB(55)
;"TDY=";Wpt(Wn,3)
4860      PRINT TAB(20);"RANGE(KM)";TAB(40);"BEARING(DEG)"
4870      PRINT "MASTER";TAB(20);Range(4);TAB(40);Bear(4)
4880      PRINT "WHISKEY";TAB(20);Range(1);TAB(40);Bear(1)
4890      PRINT "XRAY";TAB(20);Range(2);TAB(40);Bear(2)
4900      PRINT "YANKEE";TAB(20);Range(3);TAB(40);Bear(3)
4910      PRINT TAB(22);"WXY";TAB(37);"WX";TAB(52);"WY";TAB(67);"XY"
4920      PRINT "GDOP(M/NANOSEC)";TAB(20);Gdop123;TAB(35);Gdop12;TAB(50);Gdo
p13;TAB(65);Gdop23
4930      GOTO Menu
4940 Twxz:PRINT "PREDICTED: TDW=";Td(1);TAB(35);"TDX=";Td(2);TAB(55);"TDZ="
;Td(3)
4950      IF P$="N" THEN 4970
4960      PRINT "SURVEYED:  TDW=";Wpt(Wn,1);TAB(35);"TDX=";Wpt(Wn,2);TAB(55)
;"TDZ=";Wpt(Wn,4)
4970      PRINT TAB(20);"RANGE(KM)";TAB(40);"BEARING(DEG)"
4980      PRINT "MASTER";TAB(20);Range(4);TAB(40);Bear(4)
4990      PRINT "WHISKEY";TAB(20);Range(1);TAB(40);Bear(1)
5000      PRINT "XRAY";TAB(20);Range(2);TAB(40);Bear(2)
5010      PRINT "ZULU";TAB(20);Range(3);TAB(40);Bear(3)
5020      PRINT TAB(22);"WXZ";TAB(37);"WX";TAB(52);"WZ";TAB(67);"XZ"
5030      PRINT "GDOP(M/NANOSEC)";TAB(20);Gdop123;TAB(35);Gdop12;TAB(50);Gdo
p13;TAB(65);Gdop23
5040      GOTO Menu
5050 Twyz:PRINT "PREDICTED: TDW=";Td(1);TAB(35);"TDY=";Td(2);TAB(55);"TDZ="
;Td(3)
5060      IF P$="N" THEN 5080
5070      PRINT "SURVEYED:  TDW=";Wpt(Wn,1);TAB(35);"TDY=";Wpt(Wn,3);TAB(55)
;"TDZ=";Wpt(Wn,4)
5080      PRINT TAB(20);"RANGE(KM)";TAB(40);"BEARING(DEG)"
5090      PRINT "MASTER";TAB(20);Range(4);TAB(40);Bear(4)
5100      PRINT "WHISKEY";TAB(20);Range(1);TAB(40);Bear(1)
5110      PRINT "YANKEE";TAB(20);Range(2);TAB(40);Bear(2)
5120      PRINT "ZULU";TAB(20);Range(3);TAB(40);Bear(3)
5130      PRINT TAB(22);"WYZ";TAB(37);"WY";TAB(52);"WZ";TAB(67);"YZ"
5140      PRINT "GDOP(M/NANOSEC)";TAB(20);Gdop123;TAB(35);Gdop12;TAB(50);Gdo
p13;TAB(65);Gdop23
5150      GOTO Menu
5160      !
```

```
5170 K21:!  ^     CALCULATE GRID WARP
5180     PRINT PAGE
5190     PRINT " CALCULATE GRID WARP "
5200     MASS STORAGE IS ":T15"
5210     PRINTER IS 0
5220     !           INPUT CHAIN DATA
5230     IF Ci=1 THEN 5310
5250     INPUT "CHAIN FILE?",Chain$
5270     IF LEN(Chain$)<>5 THEN 5250
5280     ASSIGN #1 TO Chain$
5285     Ci=1
5290     READ #1;Xmit(*),Power(*),Emis(*)
5310     Ch$=Chain$[1,4]
5320     Conf=VAL(Chain$[5,5])
5330 ! CALCULATE CARTESIAN COORDINATES OF XMITTERS
5340     CALL Cart_coord(Xmit(*),Wpt(25,7),Wpt(25,8),Zxmit(*))
5350 ! REFERENCE WAYPOINT FOR CALCULATIONS
5360     W=Wt
5370     Zp(1)=Wpt(W,5)
5380     Zp(2)=Wpt(W,6)
5390 ! CALCULATE AND PLOT TD ERRORS
5400     ON Conf GOTO Wxyz,Wwxy,Wwxz,Wwyz
5410 Wxyz:    !
5420     T(1)=Wpt(W,2)
5430     T(2)=Wpt(W,3)
5440     T(3)=Wpt(W,4)
5450     CALL Warp(X(*),Y(*),Z(*),Zx(*),Zy(*),Zp(*),T(*),Zxmit(*),V,N,Att(*
),R,Wt,Wf,Conf,F$)
5460     GOTO Menu
5470 Wwxy:!
5480     T(1)=Wpt(W,1)
5490     T(2)=Wpt(W,2)
5500     T(3)=Wpt(W,3)
5510     CALL Warp(W(*),X(*),Y(*),Zx(*),Zy(*),Zp(*),T(*),Zxmit(*),V,N,Att(*
),R,Wt,Wf,Conf,F$)
5520     GOTO Menu
5530 Wwxz:!
5540     T(1)=Wpt(W,1)
5550     T(2)=Wpt(W,2)
5560     T(3)=Wpt(W,4)
5570     CALL Warp(W(*),X(*),Z(*),Zx(*),Zy(*),Zp(*),T(*),Zxmit(*),V,N,Att(*
),R,Wt,Wf,Conf,F$)
5580     GOTO Menu
5590 Wwyz:!
5600     T(1)=Wpt(W,1)
5610     T(2)=Wpt(W,3)
5620     T(3)=Wpt(W,4)
5630     CALL Warp(W(*),Y(*),Z(*),Zx(*),Zy(*),Zp(*),T(*),Zxmit(*),V,N,Att(*
),R,Wt,Wf,Conf,F$)
5640     GOTO Menu
5650     !
```

```
5670 K22:! ^     DAISY CHAIN
5680    PRINT PAGE
5690    PRINT " DAISY CHAIN "
5700    PRINTER IS 0
5710    MASS STORAGE IS ":T15"
5720    !       INPUT CHAIN DATA
5730    IF C1=1 THEN 5810
5750    INPUT "CHAIN FILE",Chain$
5770    IF LEN(Chain$)<>5 THEN 5750
5780    ASSIGN #1 TO Chain$
5785    C1=1
5790    READ #1;Xmit(*),Power(*),Emis(*)
5810    Conf=VAL(Chain$[5,5])
5820    Ch$=Chain$[1,4]
5830    !        INPUT WAYPOINT FILE DATA
5840    IF F1=1 THEN 5890
5850    INPUT "WP FILE?",File$
5860    F1=1
5870    ASSIGN #1 TO File$
5880    READ #1;Wpt(*)
5890    !        CALCULATE XY COORDINATES OF TRANSMITTERS
5900    CALL Cart_coord(Xmit(*),Wpt(25,7),Wpt(25,8),Zxmit(*))
5910    !        INPUT START AND STOP WAYPOINTS
5920    INPUT "START WAYPOINT NO.?",Rw
5930    INPUT "END WAYPOINT ?",Iw
5940    !        CALCULATE G-MATRICES FOR Rw
5950    FOR I=1 TO 4
5960    CALL Rb(Wpt(Rw,5),Wpt(Rw,6),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I))
5970    NEXT I
5980    Zp(1)=Wpt(Rw,5)
5990    Zp(2)=Wpt(Rw,6)
6000    CALL G_mat(Power(*),V,Range(*),Bear(*),Zp(*),Aa(*),G123(*),G12(*),G
23(*),G13(*))
6010    !        INPUT TWO OR THREE TD SOLUTION
6020    INPUT "TWO OR THREE TD SOLUTION? 2 OR 3",So
6030    IF (So<2) OR (So>3) THEN 6020
6040    IF So=2 THEN Ctwo
6050    IF So=3 THEN Cthree
6060    !        CALCULATE POSITION OF Iw
6070 Ctwo:   !       TWO TD SOLUTION
6080     CALL Wp2(Wpt(*),Conf,Rw,Tpp(*),Pair,Zpp(*),G12(*),G23(*),G13(*),G(
*),L$,Zxmit(*),Zxm(*))
6090     PRINT "TWO TD SOLUTION, CHAIN=";Ch$;"  LOPs=";L$
6100     CALL Wpf2(Wpt(*),Conf,Pair,Iw,Tqq(*))
6110     CALL Fehg(Zxm(*),Zpp(*),Tpp(*),Tqq(*),G(*),Zq(*),V,2)
6120     GOTO Print
6130 Cthree: !       THREE TD SOLUTION
6140     CALL Wp3(Wpt(*),Conf,Rw,Tp(*),Zpp(*),L$)
6150     PRINT "THREE TD SOLUTION, CHAIN=";Ch$;"  LOPs=";L$
6160     CALL Wp3(Wpt(*),Conf,Iw,Tq(*),Zxx(*),L$)
6170     CALL Fehg(Zxmit(*),Zpp(*),Tp(*),Tq(*),G123(*),Zq(*),V,3)
6180 Print:!     PRINT RESULTS
6190     DEG
6200     Dx=Zq(1,1)-Wpt(Rw,5)
```

```
6210      Dy=Zq(2,1)-Wpt(Rw,6)
6220      Dlat=Dy/(1.852*60)
6230      Dlon=Dx/(1.852*60*COS(Wpt(Rw,7)))
6240      Lat=Wpt(Rw,7)+Dlat
6250      Lon=Wpt(Rw,8)+Dlon
6260      A=ATN(Dx/Dy)
6270      IF Dy<0 THEN A=A+180
6280      IF A<0 THEN A=A+360
6290      R=SQR(Dx^2+Dy^2)
6300      FIXED 0
6310      PRINT "LOCAL GRID CENTERED AT WP";25
6320      PRINT " WAYPOINT";Iw;"REFERENCED TO";Rw
6330      FIXED 3
6340      PRINT TAB(10);"Dx=";Dx,"Dy=";Dy;TAB(40);"D_lat=";Dlat,"D_lon=";Dlo
n
6350      PRINT TAB(10);"ANGLE=";A;"   RANGE=";R;"KM(";R/1.852;"NM,";R/.9144
;"KYDS)"
6360      PRINT TAB(10);"X=";Zq(1,1),"Y=";Zq(2,1);TAB(40);"LAT=";Lat,"LON=";
Lon
6370      PRINT "CURRENT CO-ORDINATES:"
6380      PRINT TAB(10);"X=";Wpt(Iw,5),"Y=";Wpt(Iw,6);TAB(40);"LAT=";Wpt(Iw,
7),"LON=";Wpt(Iw,8)
6390      INPUT "CHANGE POSITION AND LAT/LON OF WAYPOINT OF INTEREST? Y OR N
",Ch$
6400      IF Ch$<>"Y" THEN 6460
6410      Wpt(Iw,5)=Zq(1,1)
6420      Wpt(Iw,6)=Zq(2,1)
6430      Wpt(Iw,7)=Lat
6440      Wpt(Iw,8)=Lon
6450      PRINT "WAYPOINT TABLE CHANGED"
6460      GOTO Menu
6470   !
```

```
6480 K23:! ^   WAYPOINT FILE

6490  PRINT PAGE
6500  PRINT " FILE WAYPOINT DATA "
6510  MASS STORAGE IS ":T15"
6520  CALL Wpfile(Wpt(*),F1,File$)
6530  GOTO Menu
6550  !
```

```
6560 K24:!  ^   TD MOVE
6570     PRINT PAGE
6580     PRINT " MOVE: CALCULATES Tdq=Tdp+h(Zq)-h(Zp) "
6590     PRINTER IS 0
6600     MASS STORAGE IS ":T15"
6610     FIXED 3
6620     !           INPUT CHAIN DATA
6630     IF C1=1 THEN 6710
6650     INPUT "CHAIN FILE?",Chain$
6670     IF LEN(Chain$)<>5 THEN 6650
6680     ASSIGN #1 TO Chain$
6685     C1=1
6690     READ #1;Xmit(*),Power(*),Emis(*)
6710     Ch$=Chain$[1,4]
6720     Conf=VAL(Chain$[5,5])
6730     !           INPUT WAYPOINT FILE DATA
6740     IF F1=1 THEN 6790
6750     INPUT "WAYPOINT FILE?",File$
6760     F1=1
6770     ASSIGN #1 TO File$
6780     READ #1;Wpt(*)
6790     !          INPUT WAYPOINT NUMBER AND OFFSET
6800     INPUT "WAYPOINT?",Wp
6810     INPUT "Dx/Dy1 OR Range/Bearing2, 1 OR 2",Move
6820     IF (Move<>1) AND (Move<>2) THEN 6810
6830     ON Move GOTO X_y,Rb
6840 X_y:   INPUT "INPUT Dx(KM)",Dx
6850     INPUT "INPUT Dy(KM)",Dy
6860     GOTO 6900
6870 Rb:  INPUT "RANGE(KM) AND BEARING(DEG)",R,B
6880     Dx=R*SIN(B)
6890     Dy=R*COS(B)
6900     !           CALCULATE XY COORDINATES OF TRANSMITTERS
6910     CALL Cart_coord(Xmit(*),Wpt(25,7),Wpt(25,8),Zxmit(*))
6920     !           CALCULATE h(Zp)
6930     Zp(1)=Wpt(Wp,5)
6940     Zp(2)=Wpt(Wp,6)
6950     FOR I=1 TO 4
6960     CALL Rb(Zp(1),Zp(2),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I))
6970     NEXT I
6980     FOR I=1 TO 3
6990     T(I)=(Range(I)-Range(4))/V
7000     NEXT I
7010     !           CALCULATE Dtd=h(Zq)-h(Zp)
7020     Zp(1)=Zp(1)+Dx
7030     Zp(2)=Zp(2)+Dy
7040     FOR I=1 TO 4
7050     CALL Rb(Zp(1),Zp(2),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I))
7060     NEXT I
7070     FOR I=1 TO 3
7080     T(I)=(Range(I)-Range(4))/V-T(I)
7090     NEXT I
7100     !           PRINT RESULTS AND APPLY CORRECTION TO WP(OPTIONAL)
7110     FIXED 3
```

```
7120    PRINT "AT WAYPOINT:";Wp;", A MOVE OF Dx=";Dx;"KM AND Dy=";Dy;"KM"
7130    PRINT "RESULTS IN TD CHANGES(MICROSEC) OF:"
7140    ON Conf GOTO Dxyz,Dwxy,Dwxz,Dwyz
7150 Dxyz:PRINT TAB(5);"Dtdx=";T(1);TAB(20);"Dtdy=";T(2);TAB(40);"Dtdz=";T(
3)
7160    INPUT "APPLY CORRECTION TO WP? Y OR N",Cor$
7170    IF Cor$<>"Y" THEN Menu
7180    Wpt(Wp,2)=Wpt(Wp,2)+T(1)
7190    Wpt(Wp,3)=Wpt(Wp,3)+T(2)
7200    Wpt(Wp,4)=Wpt(Wp,4)+T(3)
7210    Wpt(Wp,5)=Zp(1)
7220    Wpt(Wp,6)=Zp(2)
7240    GOTO Done
7250 Dwxy:PRINT TAB(5);"Dtdw=";T(1);TAB(20);"Dtdx=";T(2);TAB(40);"Dtdy=";T(
3)
7260    INPUT "APPLY CORRECTION TO WP? Y OR N",Cor$
7270    IF Cor$<>"Y" THEN Menu
7280    Wpt(Wp,1)=Wpt(Wp,1)+T(1)
7290    Wpt(Wp,2)=Wpt(Wp,2)+T(2)
7300    Wpt(Wp,3)=Wpt(Wp,3)+T(3)
7310    Wpt(Wp,5)=Zp(1)
7320    Wpt(Wp,6)=Zp(2)
7340    GOTO Done
7350 Dwxz:PRINT TAB(5);"Dtdw=";T(1);TAB(20);"Dtdx=";T(2);TAB(40);"Dtdz=";T(
3)
7360    INPUT "APPLY CORRECTION TO WP? Y OR N",Cor$
7370    IF Cor$<>"Y" THEN Menu
7380    Wpt(Wp,1)=Wpt(Wp,1)+T(1)
7390    Wpt(Wp,2)=Wpt(Wp,2)+T(2)
7400    Wpt(Wp,4)=Wpt(Wp,4)+T(3)
7410    Wpt(Wp,5)=Zp(1)
7420    Wpt(Wp,6)=Zp(2)
7440    GOTO Done
7450 Dwyz:PRINT TAB(5);"Dtdw=";T(1);TAB(20);"Dtdy=";T(2);TAB(40);"Dtdz=";T(
3)
7460    INPUT "APPLY CORRECTION TO WP? Y OR N",Cor$
7470    IF Cor$<>"Y" THEN Menu
7480    Wpt(Wp,1)=Wpt(Wp,1)+T(1)
7490    Wpt(Wp,3)=Wpt(Wp,3)+T(2)
7500    Wpt(Wp,4)=Wpt(Wp,4)+T(3)
7510    Wpt(Wp,5)=Zp(1)
7520    Wpt(Wp,6)=Zp(2)
7530 Done:PRINT "WAYPOINT TD AND POSITION CORRECTED"
7540    GOTO Menu
7560    !
```

```
7570 K25:1 ^   LINK DATA FILES

7580     PRINT PAGE
7590     PRINT " LINK TD DATA FILES "
7600     MAT W=(0)
7610     MAT X=(0)
7620     MAT Y=(0)
7630     MAT Z=(0)
7640     FOR I=1 TO 400
7650     U$=""
7660     NEXT I
7670     CALL Link(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
7680     GOTO Menu
```

```
7690 K26:! ^   STORE CHAIN DATA
7700    MASS STORAGE IS ":T15"
7710    PRINTER IS 16
7720    PRINT PAGE
7730    PRINT " STORE CHAIN DATA "
7733    Mask$="XYZWXYWXZWYZ"
7770    INPUT "INPUT CHAIN FILE NAME",Chain$
7775    IF (VAL(Chain$[5,5])>4) OR (VAL(Chain$[5,5])<1) OR (LEN(Chain$)<>5)
 THEN 7770
7777    Conf=VAL(Chain$[5,5])
7780    INPUT "IS THIS FILE CURRENTLY ON TAPE?",Q$
7782    IF UPC$(Q$)="N" THEN CREATE Chain$,1
7783    ASSIGN #1 TO Chain$
7784    IF UPC$(Q$)="Y" THEN READ #1;Xmit(*),Power(*),Emis(*)
7785 Op:INPUT "MODIFY/CREATE1; LIST2; SAVE3; QUIT4",Opt
7786    IF (Opt<1) OR (Opt>4) THEN Op
7787    ON Opt GOTO Cr,Lst,Sve,Menu
7810 Cr:!        INPUT MASTER DATA
7820    PRINT PAGE,LIN(20)
7821    D=M=S=0
7830    PRINT "INPUT MASTER DATA"
7840    INPUT "LATITUDE? D,M,S",D,M,S
7841    IF (D=0) AND (M=0) AND (S=0) THEN 7851
7850    Xmit(4,1)=D+M/60+S/3600
7851    D=M=S=0
7860    INPUT "LONGITUDE? D,M,S",D,M,S
7861    IF (D=0) AND (M=0) AND (S=0) THEN 7880
7870    Xmit(4,2)=-(D+M/60+S/3600)
7880    INPUT "POWER LEVEL? KW",Power(4)
7890    !        INPUT SECONDARY DATA
7891    Indx=(Conf-1)*3
7960    FOR I=1 TO 3
7970    PRINT PAGE,LIN(20)
8000    PRINT "INPUT TD";Mask$[Indx+I;1];" DATA"
8001    D=M=S=0
8010   INPUT "LATITUDE? D,M,S",D,M,S
8011   IF (D=0) AND (M=0) AND (S=0) THEN 8021
8020   Xmit(I,1)=D+M/60+S/3600
8021   D=M=S=0
8030   INPUT "LONGITUDE? D,M,S",D,M,S
8031   IF (D=0) AND (M=0) AND (S=0) THEN 8050
8040   Xmit(I,2)=-(D+M/60+S/3600)
8050   INPUT "POWER LEVEL? KW",Power(I)
8060   INPUT "EMISSION DELAY?",Emis(I)
8070   NEXT I
8071   PRINT PAGE
8072   GOTO Op
8080 Lst:  !            PRINT DATA TABLE
8081  PRINTER IS 0
8090   PRINT LIN(5);"CHAIN DATA FOR FILE: ";Chain$
8100   PRINT LIN(1);"STATION";TAB(13);"LAT";TAB(32);"LON";TAB(50);"POWER (K
W)";TAB(64);"EMISSION DELAY"
8160   IMAGE 6A,5X,DDD,AA,DD,"' ",DD.DD,"''",4X,DDD,AA,DD,"' ",DD.DD,"''",4
X,DDDD,8X,DDDDD.DD
```

```
8161   Deg$=CHR$(179)
8165   I=4
8166   GOSUB Unpak
8170   PRINT USING 8160;"MASTER",Deg1,Deg$,Min1,Sec1,Deg2,Deg$,Min2,Sec2,Po
wer(I)
8180   FOR I=1 TO 3
8181   GOSUB Unpak
8182   PRINT USING 8160;Mask$[Indx+I;1],Deg1,Deg$,Min1,Sec1,Deg2,Deg$,Min2,
Sec2,Power(I),Emis(I)
8183   NEXT I
8200   PRINT LIN(5)
8201   PRINTER IS 16
8210   GOTO Op
8220 Unpak:!
8221   Deg1=INT(Xmit(I,1))
8222   Min1=INT(FRACT(Xmit(I,1))*60)
8223   Sec1=FRACT(FRACT(Xmit(I,1))*60)*60
8224   Deg2=INT(-Xmit(I,2))
8225   Min2=INT(FRACT(-Xmit(I,2))*60)
8226   Sec2=FRACT(FRACT(-Xmit(I,2))*60)*60
8227   RETURN
8228   !
8240 Sve:!           FILE DATA
8260   ASSIGN #1 TO Chain$
8270   PRINT #1;Xmit(*),Power(*),Emis(*)
8280    GOTO Menu
8300    !
```

```
8310 K27:! ^   REFLECT TDs TO WAYPOINT
8320  MASS STORAGE IS ":T15"
8330  PRINT PAGE
8340  PRINT " REFLECT TDS TO WAYPOINT "
8350  !            INPUT CHAIN DATA
8360  IF C1=1 THEN 8440
8380  INPUT "CHAIN FILE",Chain$
8400  IF LEN(Chain$)<>5 THEN 8380
8410  ASSIGN #1 TO Chain$
8415  C1=1
8420  READ #1;Xmit(*),Power(*),Emis(*)
8440  Conf=VAL(Chain$[5,5])
8450  Ch$=Chain$[1,4]
8460  !            INPUT WAYPOINT FILE DATA
8470  IF F1=1 THEN 8520
8480  INPUT "WAYPOINT FILE NAME?",File$
8490  F1=1
8500  ASSIGN #1 TO File$
8510  READ #1;Wpt(*)
8520  !            CALCULATE XY COORDINATES OF XMITTERS
8530  CALL Cart_coord(Xmit(*),Wpt(25,7),Wpt(25,8),Zxmit(*))
8540  !            INPUT WAYPOINT OF INTEREST
8550  INPUT "WAYPOINT WHERE DATA IS TO BE REFLECTED?",W
8560  !            CALCULATE REFLECTED TD ARRAYS
8570  CALL Reflect(W,Wpt(*),W(*),X(*),Y(*),Z(*),Zx(*),Zy(*),Conf,Zxmit(*),N
,V)
8580  !            CALCULATE STATS AND PRINT RESULTS
8590  PRINTER IS 0
8600  PRINT "RESULTS OF REFLECTING FILE ";F$;"TDs TO WAYPOINT";W
8610  F$=F$&">"
8620  PRINTER IS 16
8630  GOTO K2  ! CALCULATE STATS
```

8640 K28:1 ^ CREATE REF STA FILE

```
8650  PRINT PAGE
8660  PRINT " CREATE OR READ REFERENCE STATION FILE "
8670  MASS STORAGE IS ":T15"
8680  CALL Ref_file(Ref(*),Ref$(*),Rp$)
8690  GOTO Menu
8700 END
```

```
8710 SUB Reg(Stat(*),Cov(*),S(*),R(*),Offset(*))
8720 OPTION BASE 1
8730 DEFAULT ON
8740 S(1,1)=S(1,2)=S(1,3)=Stat(3,1)+Offset(1)
8750 S(1,4)=S(1,5)=Stat(3,2)+Offset(2)
8760 S(1,6)=Stat(3,3)+Offset(3)
8770 S(2,1)=Stat(3,2)+Offset(2)
8780 S(2,2)=S(2,4)=Stat(3,3)+Offset(3)
8790 S(2,3)=S(2,5)=S(2,6)=Stat(3,4)+Offset(4)
8800 S(3,1)=S(3,2)=S(3,3)=Stat(4,1)
8810 S(3,4)=S(3,5)=Stat(4,2)
8820 S(3,6)=Stat(4,3)
8830 S(4,1)=Stat(4,2)
8840 S(4,2)=S(4,4)=Stat(4,3)
8850 S(4,3)=S(4,5)=S(4,6)=Stat(4,4)
8860 FOR I=1 TO 6
8870 Ind_var:!
8880 Iv=1
8890 D=2
8900 IF S(4,I)>S(3,I) THEN Iv=2
8910 IF S(4,I)>S(3,I) THEN D=1
8920 R(3,I)=Iv
8930 Slope:!
8940 R(1,I)=Cov(3,I)*S(D+2,I)/S(Iv+2,I)
8950 IF Iv=2 THEN R(1,I)=1/R(1,I)
8960 Residual:!
8970 C=1-Cov(3,I)^2
8980 IF C<0 THEN 9000
8990 R(2,I)=S(D+2,I)*C^.5
9000 NEXT I
9010 SUBEND
9030  !
```

```
9040 SUB Print(Stats(*),O_set(*),Cov(*),R(*),Sample,V)
9050  OPTION BASE 1
9051  DIM Bar$[80]
9052 Bar$="________________________________________________________________
_______________"
9060  FOR I=1 TO 3
9070  FOR J=1 TO 6
9080  IF R(I,J)>100 THEN R(I,J)=0
9090  IF Cov(3,J)>100 THEN Cov(3,J)=0
9100  NEXT J
9110  NEXT I
9121   PRINT Bar$
9130   IF V=2 THEN 9280
9140 PRINT LIN(1);TAB(21);" TDW ";TAB(36);" TDX ";TAB(51);" TDY ";TAB(66);"
 TDZ "
9150   FIXED 3
9160   PRINT LIN(1);TAB(1);"CUMULATIVE AVERAGE";TAB(20);Stats(3,1)+O_set(1)
;TAB(35);Stats(3,2)+O_set(2);TAB(50);Stats(3,3)+O_set(3);TAB(65);Stats(3,4)
+O_set(4)
9170   PRINT LIN(1);TAB(1);"STANDARD DEVIATION";TAB(20);Stats(4,1);TAB(35);
Stats(4,2);TAB(50);Stats(4,3);TAB(65);Stats(4,4)
9180   PRINT LIN(1);TAB(1);"TD PAIR";TAB(17);"WX";TAB(27);"WY";TAB(37);"WZ"
;TAB(47);"XY";TAB(57);"XZ";TAB(67);"YZ"
9190   PRINT LIN(1);TAB(1);"CORR COEF";TAB(15);Cov(3,1);TAB(25);Cov(3,2);TA
B(35);Cov(3,3);TAB(45);Cov(3,4);TAB(55);Cov(3,5);TAB(65);Cov(3,6)
9200   PRINT LIN(1);TAB(1);"SLOPE";TAB(15);R(1,1);TAB(25);R(1,2);TAB(35);R(
1,3);TAB(45);R(1,4);TAB(55);R(1,5);TAB(65);R(1,6)
9210   PRINT LIN(1);TAB(1);"RESIDUAL";TAB(15);R(2,1);TAB(25);R(2,2);TAB(35)
;R(2,3);TAB(45);R(2,4);TAB(55);R(2,5);TAB(65);R(2,6)
9220   STANDARD
9230   PRINT LIN(1);TAB(1);"IND VAR";TAB(17);R(3,1);TAB(27);R(3,2);TAB(37);
R(3,3);TAB(47);R(3,4);TAB(57);R(3,5);TAB(67);R(3,6)
9240   PRINT LIN(1);"SAMPLES=";Sample
9250   PRINT Bar$
9260   PRINT LIN(1)
9270   SUBEXIT
9280   FIXED 3
9290   PRINT LIN(1);TAB(21);" R1  ";TAB(36);" R2  ";TAB(51);" X   ";TAB(66)
;" Y   "
9300   PRINT LIN(1);TAB(1);"CUMULATIVE AVERAGE";TAB(20);Stats(3,1)+O_set(1)
;TAB(35);Stats(3,2)+O_set(2);TAB(50);Stats(3,3)+O_set(3);TAB(65);Stats(3,4)
+O_set(4)
9310   PRINT LIN(1);TAB(1);"STANDARD DEVIATION";TAB(20);Stats(4,1);TAB(35);
Stats(4,2);TAB(50);Stats(4,3);TAB(65);Stats(4,4)
9320   PRINT Bar$
9330   SUBEND
9350   !
```

```
9360 SUB Hi_lo(X(*),N,Hi,Lo)
9370 Hi=X(1)
9380 Lo=X(1)
9390 FOR I=1 TO N
9400 Lo=MIN(X(I),Lo)
9410 Hi=MAX(X(I),Hi)
9420 NEXT I
9430 SUBEND
```

```
9440 SUB Plot(Lx,Hx,Ly,Hy,Xx,Yx,X(*),Y(*),S1,N,X$,Y$,F$)
9450 OPTION BASE 1
9460 PLOTTER IS 13,"GRAPHICS"
9470 GRAPHICS
9480 LOCATE 10,90,20,100
9490 SHOW Lx-1,Hx+1,Ly-1,Hy+1
9500 AXES 1,1,Xx,Yx
9510 LINE TYPE 2
9520 FOR I=1 TO N
9530 DRAW X(I),Y(I)
9540 NEXT I
9550 LINE TYPE 1
9560 MOVE Lx,S1*(Lx-Xx)+Yx
9570 DRAW Hx,S1*(Hx-Xx)+Yx
9580 Label:CSIZE 3
9590  LORG 5
9600  LOCATE 0,125,0,20
9610  SCALE 0,100,0,20
9620  MOVE 50,15
9630   LABEL USING "#,K";X$;" VS ";Y$;" & REGRESSION LINE"
9640  MOVE 50,10
9650 LABEL USING "K";"ONE MICROSEC/DIV"
9660  MOVE 50,5
9670  LABEL USING "K";"T[illegible]LE="&F$
9680  PAUSE
9690  INPUT "HARD COPY? Y OR N",Hc$
9700  IF Hc$<>"Y" THEN 9720
9710 DUMP GRAPHICS
9720  EXIT GRAPHICS
9730 SUBEND
9750  !
```

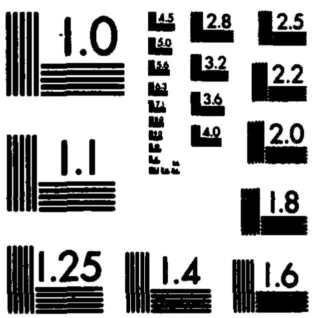
```
9760 SUB Rplot(Iv(*),D(*),S(*),R(*),Pr,N,V1$,V2$,F$)
9770 OPTION BASE 1
9780  P1=1
9790  PRINTER IS 0
9800  INPUT "PLOT RESIDUALS VS N(1) OR INDEPENDENT VAR(2)",P1
9810 PLOTTER IS 13,"GRAPHICS"
9820 GRAPHICS
9830 LOCATE 10,100,20,80
9840  IF P1=1 THEN 9890
9850  CALL Hi_lo(Iv(*),N,Imax,Imin)
9860  SCALE Imin-2,Imax,-5,5
9870  AXES 1,1,Imin,0
9880  GOTO 9910
9890 SCALE 0,N,-5,5
9900 AXES 10,1,0,0
9910 LINE TYPE 2
9920  R$=VAL$(R(2,Pr))
9930 M=R(1,Pr)
9940 IF R(3,Pr)=2 THEN M=1/M
9950 Iv=1
9960 D=2
9970 IF R(3,Pr)=2 THEN Iv=2
9980 IF R(3,Pr)=2 THEN D=1
9990 Da=S(D,Pr)
10000 Ia=S(Iv,Pr)
10010 FOR I=1 TO N
10020 Dp=Da+M*(Iv(I)-Ia)
10030 Dr=Dp-D(I)
10040 Nr=Dr/R(2,Pr)
10050 IF Nr>5 THEN PRINT I,Nr
10060  IF P1=1 THEN DRAW I,Nr
10070  IF P1=2 THEN DRAW Iv(I),Nr
10100 NEXT I
10110 PRINTER IS 16
10120 LINE TYPE 1
10130 Ctr=N/8
10140 IF P1=2 THEN Ctr=Imin+(Imax-Imin)/8
10150  MOVE Ctr,-4
10151  IF P1=1 THEN LABEL USING "K";"NORMALIZED RESIDUALS VS SAMPLE NUMBER"
10160 IF P1=2 THEN LABEL USING "K";"NORMALIZED RESIDUALS VS INDEP VAR"
10170 MOVE Ctr,-5
10180 LABEL USING "K";V1$;" AND ";V2$;" DATA";"; RES=";R$;" MICROSEC"
10190 MOVE Ctr,-6
10200 LABEL USING "K";"TRACKLINE=";F$
10201 IF P1=1 THEN 10320
10210 MOVE Ctr,4
10220 LABEL USING "K";Imin
10230 MOVE Ctr,5
10240 LABEL USING "K";"IND VAR MIN"
10320 INTEGER K
10321 LORG 5
10330 IF P1=1 THEN 10360
10340 K=Imax-Imin
10350 GOTO 10390
10360 K=N/10
```

```
10370 IF (N>100) AND (P1=1) THEN K=N/50
10380 Step=1
10390 IF (P1=2) AND (K>10) THEN Step=5
10400 FOR I=0 TO K STEP Step
10401 IF I=0 THEN 10480
10410 Mx=Imin+I
10420 IF P1=1 THEN Mx=10*I
10430 IF (N>100) AND (P1=1) THEN Mx=50*I
10440 MOVE Mx,-1
10450 IF P1=1 THEN L=Mx
10460 IF P1=2 THEN L=I
10470 LABEL USING "K";L
10480 NEXT I
10490 IF P1=2 THEN 10580
10500 FOR I=1 TO 5
10510 M=N
10520 MOVE -M/16,I
10530 LABEL USING "K";I
10540 MOVE -M/16,-I
10550 LABEL USING "K";-I
10560 NEXT I
10570 GOTO 10640
10580 FOR I=1 TO 5
10590 MOVE Imin-3,I
10600 LABEL USING "K";I
10610 MOVE Imin-3,-I
10620 LABEL USING "K";-I
10630 NEXT I
10640 PAUSE
10650 INPUT "HARD COPY? Y OR N",Hc$
10660 IF Hc$="Y" THEN DUMP GRAPHICS
10670 EXIT GRAPHICS
10680 SUBEND
10700 !
```

AD-A124 343 HHE/LORAN-C SURVEYING(U) COAST GUARD WASHINGTON DC OFFICE OF RESEARCH AND DEVELOPMENT A J SEDLOCK NOV 82 USCG-D-54-82 4/4

UNCLASSIFIED F/G 17/7 NL

END
DATE
FILMED
83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
10710 SUB Bear(P1,P2,P3,P4,B)
10720 ! PROGRAM TO CALCULATE BEARING BETWEEN TWO POINTS
10730 ! (P1,P2) AND (P3,P4).POSITIONS ARE INPUT IN DEGREES
10740 ! OF LAT/LON . N AND E ARE +; S AND W ARE -
10750 DEG
10760 P5=COS(P3)
10770 P6=P4-P2
10780 P7=SIN(P3)*COS(P1)-P5*SIN(P1)*COS(P6)
10790 B=ATN(P5*SIN(P6)/P7)
10800 IF P7<0 THEN B=B+180
10810 IF B<0 THEN B=B+360
10820 SUBEND
10850 !
```

```
10860 SUB Range(P1,P2,P3,P4,R)
10870 ! PROGRAM TO CALCULATE RANGE IN KILOMETERS BETWEEN TWO
10880 ! POINTS (P1,P2) AND (P3,P4). POSITIONS ARE IN DEGREES
10890 ! (LAT,LON) . NORTH AND EAST ARE +.
10900 DEG
10910 P6=SIN(P1)
10920 P7=SIN(P3)
10930 P8=P6*P7+COS(P1)*COS(P3)*COS(P4-P2)
10940 RAD
10950 P9=(1-P8^2)^.5
10960 P5=ATN(P9/P8)
10970 P10=(P6+P7)^2*(3*P9-P5)/(1+P8)
10980 P11=(P6-P7)^2*(3*P9+P5)/(1-P8)
10990 DEG
11000 R=6378.135*P5+5.346125*(P10-P11)
11010 SUBEND
11040 !
```

```
11150 SUB G_mat(P(*),V,Range(*),Bear(*),Zp(*),A(*),G123(*),G12(*),G23(*),G1
3(*))
11160 ! PROGRAM CALCULATES:
11170 !          A(*);GRADIENT MATRIX(XY TO TD)
11180 !          G123(*);GRADIENT MATRIX (TD TO XY),3-TD CASE
11190 !          G12(*),G23(*),G13(*);GRADIENT MATRICES(TD TO XY),2-TD CASES
11200 !        INPUTS ARE:
11210 !          P(*);POWER LEVEL OF XMITTERS (KILOWATTS)
11220 !          V;VELOCITY OF PROPOGATION (MICROSEC/KILOMETER)
11230 !          Range(*);RANGES FROM POINT TO XMITTERS (KILOMETERS)
11240 !          Bear(*);AZIMUTH OF XMITTERS FROM POINT (DEGREES)
11250 !          Zp(*);XY POSITION OF POINT
11260 DEG
11270 OPTION BASE 1
11280 DIM R(3,3),W(3,3),B(2,3),C(2,2),D(2,3)
11290 MAT R=(1)
11300 K=P(4)/Range(4)^2
11310 FOR I=1 TO 3
11320 ! CALCULATE A MATRIX
11330 A(I,1)=(SIN(Bear(4))-SIN(Bear(I)))/V
11340 A(I,2)=(COS(Bear(4))-COS(Bear(I)))/V
11350 ! CALCULATE COVARIANCE MATRIX
11360 R(I,I)=1+K*Range(I)^2/P(I)
11370 NEXT I
11380 ! CALCULATE WEIGHTING MATRIX
11390 MAT W=INV(R)
11400 ! CALCULATE G123
11410 MAT B=TRN(A)
11420 MAT D=B*W
11430 MAT C=D*A
11440 MAT C=INV(C)
11450 MAT G123=C*D
11460 ! CALCULATE G12,G23,G13
11470 FOR I=1 TO 2
11480 FOR J=1 TO 2
11490 K=I+1
11500 L=J+1
11510 M=I
11520 N=J
11530 IF I=2 THEN M=3
11540 IF J=2 THEN N=3
11550 G12(I,J)=A(I,J)
11560 G23(I,J)=A(K,J)
11570 G13(I,J)=A(M,J)
11580 NEXT J
11590 NEXT I
11600 MAT G12=INV(G12)
11610 MAT G23=INV(G23)
11620 MAT G13=INV(G13)
11630 SUBEND
11640 !
```

```
11670 SUB Td(R(*),V,Ed(*),T(*))
11680 ! PROGRAM PREDICTS TDS FOR A POINT. INPUTS ARE RANGES
11690 ! TO THE XMITTERS(R(*)),EMISSION DELAYS(Ed(*)),AND
11700 ! VELOCITY OF PROPOGATION
11710 !
11720 ! CALCULATE SECONDARY PHASE FACTORS (SF)
11730 FOR I=1 TO 4
11750 S(I)=38.4488/R(I)-.40758+.002166735*R(I)
11770 IF R(I)<160 THEN S(I)=.816768/R(I)-.011402+.0011*R(I)
11780 NEXT I
11790 ! CALCULATE TDS
11800 FOR I=1 TO 3
11810 T(I)=(R(I)-R(4))/V+S(I)-S(4)+Ed(I)
11820 NEXT I
11830 SUBEND
```

```
12030  SUB Wpfile(W(*),F1,File$)
12040 ! PROGRAM TO STORE WAYPOINT INFO
12050      OPTION BASE 1
12060      PRINTER IS 0
12070      IF F1=1 THEN Opt
12080      INPUT "WAYPOINT FILE NAME?",File$
12090      F1=1
12100      INPUT "DOES THIS FILE CURRENTLY EXIST ON TAPE? Y OR N",Cr$
12110      IF UPC$(Cr$[1,1])="N" THEN CREATE File$,8
12130 Cont: ASSIGN #1 TO File$
12140      IF UPC$(Cr$[1,1])="Y" THEN READ #1;W(*)
12141 Opt:  INPUT "MODIFY/CREATE1; LIST2; SAVE3; QUIT4",Opt
12142      IF (Opt<1) OR (Opt>4) THEN Opt
12145      ON Opt GOTO Input,List,File,End
12160 Input:INPUT "WAYPOINT NUMBER?",Nw
12165      IF (Nw<1) OR (Nw>25) THEN Input
12170      PRINT "WAYPOINT NUMBER=";Nw
12180      INPUT "INPUT OR CHANGE TDs? Y OR N",Td$
12190      IF Td$="N" THEN Xy
12200      INPUT "TDW,TDX,TDY,TDZ ?",W(Nw,1),W(Nw,2),W(Nw,3),W(Nw,4)
12210      FIXED 2
12220      PRINT "TDW=";W(Nw,1),"TDX=";W(Nw,2),"TDY=";W(Nw,3),"TDZ=";W(Nw,4
)
12230 Xy:   INPUT "INPUT OR CHANGE XY POSITION? Y OR N",P$
12240 IF P$="N" THEN L1
12250 FIXED 3
12260 INPUT "XY POSITION? E,N",W(Nw,5),W(Nw,6)
12270 PRINT "X=";W(Nw,5),"Y=";W(Nw,6)
12280 L1:INPUT "INPUT OR CHANGE LAT/LON? Y OR N",L1$
12290 IF L1$="N" THEN Next
12300 FIXED 4
12301 INPUT "ENTER LATITUDE; D,M,S",Deg,Min,Sec
12302 W(Nw,7)=Deg+Min/60+Sec/3600
12310 INPUT "ENTER LONGITUDE; D,M,S",Deg,Min,Sec
12311 W(Nw,8)=-(Deg+Min/60+Sec/3600)
12320 PRINT "LAT=";W(Nw,7),"LON=";W(Nw,8)
12330 Next:INPUT "ANOTHER WAYPOINT? Y OR N",A$
12340 IF A$="Y" THEN 12160
12350 GOTO Opt
12351 List: PRINT LIN(5);TAB(25);"WAYPOINT FILE: ";File$
12370      FOR I=1 TO 25
12380      FIXED 0
12400      PRINT LIN(1);"WAYPOINT=";I
12410      FIXED 2
12420      PRINT "TDW=";W(I,1),"TDX=";W(I,2),"TDY=";W(I,3),"TDZ=";W(I,4)
12430      FIXED 3
12440      PRINT "X=";W(I,5),"Y=";W(I,6)
12450      FIXED 0
12451      Lat=W(I,7)
12452      Deg=INT(Lat)
12453      Min1=60*(Lat-Deg)
12454      Min=INT(Min1)
12455      Sec=60*(Min1-Min)
12456      PRINT "LAT:";Deg;CHR$(179);Min;"'";
12457      FIXED 2
```

```
12458      PRINT Sec;"''    ";
12459      FIXED 0
12460      Lon=-W(I,8)
12461      Deg=INT(Lon)
12462      Mini=60*(Lon-Deg)
12463      Min=INT(Mini)
12464      Sec=60*(Mini-Min)
12465      PRINT "LON:";Deg;CHR$(179);Min;"'";
12466      FIXED 2
12467      PRINT Sec;"''"
12470      NEXT I
12471      PRINT LIN(5)
12475      GOTO Opt
12480 File:ASSIGN #1 TO File$
12485      PRINT #1;W(*)
12520 End:SUBEND
12540 !
```

```
12550 SUB Fehg(X(*),Zp(*),Tp(*),Tq(*),G(*),Zq(*),V,Nsta)
12560 ! THIS SUBROUTINE CALCULATES POSITION COORDINATES,Zq(2,1),
12570 ! USING THE FLAT EARTH HYPERBOLIC GRID (FEGH) ALGROTHIM.
12580 ! INPUTS ARE TRANSMITTER POSITIONS,X(2,3),WAYPOINT POSITION,
12590 ! Zp(2,1); WAYPOINT TDs, Tp(2,1); THE GRADIENT MATRIX, G(2,2);
12600 ! THE VELOCITY OF PROPOGATION, V; AND THE MEASURED TDs, Tq(2,1).
12610 OPTION BASE 1
12620 DIM A(Nsta,1),B(Nsta,1),C(Nsta,1),D(Nsta,1),E(2,1),T(Nsta,1),Z(2,1),R
(Nsta+1,1)
12630 Initialize:!
12640 K=0
12650 MAT Zq=Zp
12660 MAT B=Zp
12670 MAT A=Zp
12680 MAT C=Tq
12690 MAT D=Tp
12700 ! CALCULATE Zq(1)=Zp+G(Tq-Tp)
12710 MAT T=C-D
12720 MAT Zq=G*T
12730 MAT Zq=Zp+Zq
12740 ! CALULATE h(Zq(0))=h(Zp)
12750 FOR I=1 TO Nsta+1
12760 R(I,1)=SQR((Zp(1,1)-X(I,1))^2+(Zp(2,1)-X(I,2))^2)
12770 NEXT I
12780 FOR I=1 TO Nsta
12790 D(I,1)=(R(I,1)-R(Nsta+1,1))/V
12800 NEXT I
12810 ! CALCULATE Zq(n)=2Zq(n-1)-Zq(n-2)+G(h(Zq(n-2)-h(Zq(n-1)))
12820 Iterate:K=K+1
12830 DISP K
12840 ! IF ITERATIONS EXCEED 20 THEN STOP
12850 IF K>20 THEN Error
12860 MAT B=A ! Zq(n-2)=Zq(n-1)
12870 MAT A=Zq! Zq(n-1)=Zq(n)
12880 MAT C=D ! h(Zq(n-2))=h(Zq(n-1))
12890 ! CALCULATE h(Zq(n-1))
12900 FOR I=1 TO Nsta+1
12910 R(I,1)=SQR((Zq(1,1)-X(I,1))^2+(Zq(2,1)-X(I,2))^2)
12920 NEXT I
12930 FOR I=1 TO Nsta
12940 D(I,1)=(R(I,1)-R(Nsta+1,1))/V
12950 NEXT I
12960 MAT Zq=Zq*(2)
12970 MAT Zq=Zq-B
12980 MAT T=C-D
12990 MAT Z=G*T
13000 MAT Zq=Zq+Z
13010 ! CALCULATE ABS(Zq(n)-Zq(n-1))
13020 MAT E=A-Zq
13030 Diff=E(1,1)^2+E(2,1)^2
13040 ! IF DIFFERENCE >1 YARD THEN ITERATE
13050 IF SQR(Diff)>.0010 THEN Iterate
13060 DISP K
13070 GOTO End
13080 Error:DISP "20 ITERATIONS WITHOUT CLOSURE !!!!!"
```

```
13090 PAUSE
13100 End:SUBEND
13120 !
```

```
13130 SUB Xplot(T,Ax(*),Ay(*),Minx,Maxx,Miny,Maxy,N,No,F$,Ch$,L$,P1)
13140 DIM L1$[80],L4$[80]
13150 FIXED 3
13160 Zoom$=""
13170 IF No=2 THEN 13260
13180 PLOTTER IS 13,"GRAPHICS"
13190 GRAPHICS
13200 LOCATE 0,100,20,100
13210 FRAME
13220  IF Zoom$<>"Y" THEN SHOW Minx-.1,Maxx+.1,Miny-.1,Maxy+.1
13230  IF Zoom$="Y" THEN SHOW Minx,Maxx,Miny,Maxy
13240 AXES .1,.1,Minx,Miny,10,10
13250 GRID 2,2,Minx,Miny,1,1
13260 LINE TYPE T
13270 MOVE Ax(1),Ay(1)
13280 FOR I=1 TO N
13290 DRAW Ax(I),Ay(I)
13300 NEXT I
13310 PRINTER IS 0
13320 P=0
13330 IF (P1=3) AND (No=1) THEN SUBEXIT
13340 IF (P1=3) AND (No=2) THEN Label
13350 PAUSE
13360 INPUT "ZOOM?,Y OR N",Zoom$
13370 IF Zoom$<>"Y" THEN 13410
13380 DIGITIZE Minx,Miny
13390 DIGITIZE Maxx,Maxy
13400 GOTO 13180
13410 INPUT "FIND SAMPLE NUMBER AND VALUE OF PLOTTED DATA POINT? Y OR N",Ou
t$
13420 IF Out$<>"Y" THEN 13520
13430 DIGITIZE X,Y
13440 Tol=.1
13450 IF ABS(Maxx-Minx)<=1 THEN Tol=.010
13460 FOR I=1 TO N
13470 IF (ABS(Ax(I)-X)<Tol) AND (ABS(Ay(I)-Y)<Tol) THEN P=1
13480 IF P THEN PRINT I,Ax(I),Ay(I)
13490 P=0
13500 NEXT I
13510 GOTO 13410
13520 INPUT "DIGITIZE?",D$
13530 IF D$<>"Y" THEN Label
13540 INPUT "HOW MANY?<=10",P
13550 IF P>10 THEN P=10
13560 GRAPHICS
13570 FOR I=1 TO P
13580 DIGITIZE Xx(I),Yy(I)
13590 LORG 5
13600 MOVE Xx(I),Yy(I)
13610 LABEL USING "K";"+"
13620 MOVE Xx(I),Yy(I)
13630 LORG 2
13640 LABEL USING "3D";I
13650 NEXT I
13660 FOR I=1 TO P
```

```
13670 PRINT I,Xx(I),Yy(I)
13680 NEXT I
13690 Label:IF (P1=3) AND (No=1) THEN SUBEXIT
13700 LOCATE 0,100,0,20
13710 SCALE 0,100,0,25
13720 L1$="AXES: X="&VAL$(Minx)&"  Y="&VAL$(Miny)
13730 L2$="UNITS: 1 KM/DIV"
13740 L3$="FILE="&F$
13745 L4$="LORAN-C DATA:"&"  CHAIN="&Ch$&", LOPs="&L$
13790 IF P1=1 THEN L4$="MINI-RANGER DATA"
13810 IF P1=3 THEN L4$="MINI-RANGER (SOLID LINE) AND LORAN-C (DASHED LINE)"
13820 MOVE 10,20
13830 LABEL L1$
13840 MOVE 10,15
13850 LABEL L2$
13860 MOVE 10,10
13870 LABEL L3$
13880 MOVE 10,5
13890 LABEL L4$
13900 PAUSE
13910 Hrd_cop:INPUT "HARD COPY OF GRAPHICS?",H$
13920 IF H$="Y" THEN DUMP GRAPHICS
13930 EXIT GRAPHICS
13940 SUBEND
13960 !
```

```
13970 SUB Ct_at(X(*),Y(*),Ct(*),At(*),Wx1,Wy1,Wx2,Wy2,N,R,A)
13980 DEFAULT ON
13990 OPTION BASE 1
14000 DEG
14010 MAT Ct=(0)
14020 MAT At=(0)
14030 ! CALCULATE COURSE ANGLE FROM W1 TO W2
14040 Dy=Wy2-Wy1
14050 Dx=Wx2-Wx1
14060 R=SQR(Dx^2+Dy^2)
14070 A=ATN(Dx/Dy)
14080 IF Dy<0 THEN A=A+180
14090 IF A<0 THEN A=A+360
14100 ! CALCULATE ALONG TRACK AND CROSS TRACK DISTANCE
14110 FOR I=1 TO N
14120 X=X(I)-Wx1
14130 Y=Y(I)-Wy1
14140 At(I)=X*SIN(A)+Y*COS(A)
14150 Ct(I)=-X*COS(A)+Y*SIN(A)
14160 NEXT I
14170 ! FILTER Ct
14180   FOR I=2 TO N
14190   Ct(I)=Ct(I-1)*.7+Ct(I)*.3
14200   NEXT I
14210 SUBEND
14230 !
```

```
14240 SUB Patct(At(*),Ct(*),N,Mina,Maxa,Minc,Maxc,Wt,Wf,F$,R,No,Data)
14241 DEG
14250 DIM L1$[40],L2$[40],L3$[40],L4$[80]
14260 IF No=2 THEN 14370
14270 PLOTTER IS 13,"GRAPHICS"
14280 GRAPHICS
14290 LOCATE 10,100,20,100
14300 Mina=MIN(0,Mina)
14310 Maxa=MAX(R,Maxa)
14330 Bc=MAX(Maxc,.22)
14340 Lc=MIN(Minc,-.22)
14350 SCALE Mina,Maxa,Lc,Bc
14360 AXES .1,.01,0,0,10,10
14370 MOVE At(1),Ct(1)
14380 LINE TYPE 1
14390 IF (Data=3) AND (No=2) THEN LINE TYPE 2
14400 FOR I=1 TO N
14410 DRAW At(I),Ct(I)
14420 NEXT I
14430 MOVE R,0
14440 LORG 5
14450 LABEL USING "K";"I"
14460 LINE TYPE 1
14470 IF (Data=3) AND (No=1) THEN SUBEXIT
14480 Label:!
14490 Y_axis:!
14500 Y=-R/30
14501 LORG 8
14510 ! IF Maxa>5 THEN Y=.5
14520 MOVE Y,.1
14530 LABEL USING "K";100
14540 MOVE Y,.2
14550 LABEL USING "K";200
14560 MOVE Y,-.1
14570 LABEL USING "K";-100
14580 MOVE Y,-.2
14590 LABEL USING "K";-200
14600 MOVE Y,-.215
14601 LORG 5
14602 LDIR 270
14603 MOVE Y,0
14610 LABEL USING "K";"METERS"
14611 LDIR 0
14620 X_axis:!
14630 INTEGER K
14640 K=Maxa
14650 FOR I=1 TO K
14660 MOVE I,-.015
14670 LABEL USING "K";I
14680 NFXT I
14690 MOVE K-1,-.035
14700 LABEL USING "K";"KILOMETERS"
14710 LOCATE 10,100,0,20
14720 SCALE 0,100,0,25
14730 FIXED 0
```

```
14740 L1$="ALONG TRACK VS CROSS TRACK "
14750 L2$="WAYPOINT "&VAL$(Wf)&" TO "&"WAYPOINT "&VAL$(Wt)
14760 L3$="FILE="&F$
14770 L4$="MINIRANGER DATA"
14780 IF Data=2 THEN L4$="LORAN-C DATA"
14790 IF Data=3 THEN L4$="LORAN-C DATA(DOTTED LINE) AND MINIRANGER DATA(SOL
ID LINE)"
14800 MOVE 0,22
14810 LABEL "WP"&VAL$(Wt)
14820 MOVE 100,22
14830 LABEL "WP"&VAL$(Wf)
14840 MOVE 55,20
14850 LABEL L1$
14860 MOVE 55,15
14870 LABEL L2$
14880 MOVE 55,10
14890 LABEL L3$
14900 MOVE 55,5
14910 LABEL L4$
14920 PAUSE
14930 INPUT "HARD COPY? Y OR N",Hc$
14940 IF Hc$="Y" THEN DUMP GRAPHICS
14950 EXIT GRAPHICS
14960 SUBEND
14980 !
```

```
14990 SUB Delete(W(*),X(*),Y(*),Z(*),R1(*),R2(*),T$(*),N,F$)
15000 OPTION BASE 1
15010 DIM We(400),Xe(400),Ye(400),Ze(400),Te$(400),R1e(400),R2e(400)
15020 PRINTER IS 0
15030 Sample:INPUT "SAMPLE TO BE DELETED?START WITH HIGHEST NUMBER ",D
15040 J=0
15050 FOR I=1 TO N
15060 Sampl:IF I=D THEN Next
15070 J=J+1
15080 DISP I,J
15090 We(J)=W(I)
15100 Xe(J)=X(I)
15110 Ye(J)=Y(I)
15120 Ze(J)=Z(I)
15130 R1e(J)=R1(I)
15140 R2e(J)=R2(I)
15150 Te$(J)=T$(I)
15160 Next: NEXT I
15170 N=J
15180 MAT W=We
15190 MAT X=Xe
15200 MAT Y=Ye
15210 MAT Z=Ze
15220 MAT R1=R1e
15230 MAT R2=R2e
15240 FOR I=1 TO N
15250 T$(I)=Te$(I)
15260 NEXT I
15270 PRINT "SAMPLE";D;"DELETED,";N;"SAMPLES REMAIN"
15280 INPUT "ANOTHER SAMPLE TO DELETE?",A$
15290 IF A$="Y" THEN Sample
15370 End:SUBEND
15390 !
```

```
15400 SUB Read(W(*),X(*),Y(*),Z(*),R1(*),R2(*),Time$(*),Sample,F$)
15410 OPTION BASE 1
15420 DIM A(100),B(100),C(100),D(100),T$(100),E(100),F(100)
15430 MASS STORAGE IS ":T14"
15440 Sample=0                  ! ZERO SAMPLE NUMBER AND DATA ARRAYS
15450 MAT W=(0)
15460 MAT X=(0)
15470 MAT Y=(0)
15480 MAT Z=(0)
15490 MAT R1=(0)
15500 MAT R2=(0)
15510 FOR I=1 TO 400
15520 Time$(I)=""
15530 NEXT I
15540 INPUT "  FILE NAME? ",F$    ! INPUT FILE NAME
15550 ASSIGN #1 TO F$
15560   READ #1;Set,Sample        ! READ NUMBER OF DATA SETS AND SAMPLES
15570 FOR J=0 TO 3                ! READ DATA SET BY SET AND LOAD INTO
15580 ON END #1 GOTO 15600        ! DATA ARRAYS
15590 READ #1;A(*),B(*),C(*),D(*),E(*),F(*),T$(*)
15600 FOR I=1 TO 100
15610 W(I+J*100)=A(I)
15620 X(I+J*100)=B(I)
15630 Y(I+J*100)=C(I)
15640 Z(I+J*100)=D(I)
15650 R1(I+J*100)=E(I)
15660 R2(I+J*100)=F(I)
15670 Time$(I+J*100)=T$(I)
15680 IF I+J*100=Sample THEN SUBEXIT
15690 NEXT I
15700   IF J+1=Set THEN SUBEXIT
15710 NEXT J
15720 SUBEND
15730 !
```

```
15750 SUB Link(W(*),X(*),Y(*),Z(*),R1(*),R2(*),Time$(*),Count,L$)
15760 OPTION BASE 1
15770 DIM A(100),B(100),C(100),D(100),T$(100),E(100),F(100)
15780 MASS STORAGE IS ":T14"
15790 Count=0
15800 INPUT "FIRST FILE NAME? ",F$
15810 L$=""
15811 Comma$=""
15820 Assign:ASSIGN #1 TO F$
15830   READ #1;Set,N
15840 FOR J=0 TO 3
15850 ON END #1 GOTO 15870
15860 READ #1;A(*),B(*),C(*),D(*),E(*),F(*),T$(*)
15870 FOR I=1 TO 100
15880 Kount=Count+I+J*100
15890 DISP Kount
15900 W(Kount)=A(I)
15910 X(Kount)=B(I)
15920 Y(Kount)=C(I)
15930 Z(Kount)=D(I)
15940 R1(Kount)=E(I)
15950 R2(Kount)=F(I)
15960 Time$(Kount)=T$(I)
15970 IF I+J*100=N THEN Jump
15990 IF Kount=400 THEN Jump
16000 NEXT I
16010 IF Set=J+1 THEN Jump
16020 NEXT J
16030 Jump:!
16040 PRINTER IS 16
16050 PRINT PAGE;LIN(20);"CURRENT FILE=";F$
16060 INPUT "CORRECT RANGE DATA? Y OR N",C$
16070 PRINTER IS 0
16080 IF C$<>"Y" THEN 16180
16090 F$=F$&"r"
16100 INPUT "CORRECTION TO R1(METERS)?",R1c
16110 INPUT "CORRECTION TO R2(METERS)?",R2c
16120 PRINT "FILE=";F$
16130 PRINT "R1 CORRECTION=";R1c;TAB(30);"R2 CORRECTION=";R2c
16140 FOR I=Count+1 TO Kount
16150 R1(I)=R1(I)+R1c
16160 R2(I)=R2(I)+R2c
16170 NEXT I
16180 INPUT "CORRECT TD DATA? Y OR N",C$
16190 IF C$<>"Y" THEN 16340
16200 PRINT "START TIME=";Time$(Count+1);TAB(30);"STOP TIME=";Time$(Kount)
16210 INPUT "CORRECTION TO TDW(MICROSEC)?",Wc
16220 INPUT "CORRECTION TO TDX(MICROSEC)?",Xc
16230 INPUT "CORRECTION TO TDY(MICROSEC)?",Yc
16240 INPUT "CORRECTION TO TDZ(MICROSEC)?",Zc
16250 FOR I=Count+1 TO Kount
16260 W(I)=W(I)+Wc
16270 X(I)=X(I)+Xc
16280 Y(I)=Y(I)+Yc
16290 Z(I)=Z(I)+Zc
```

```
16300 NEXT I
16310 F$=F$&"t"
16320 PRINT "FILE=";F$
16330 PRINT "TD CORRECTIONS:";TAB(20);"Wcor=";Wc;TAB(35);"Xcor=";Xc;TAB(50)
;"Ycor=";Yc;TAB(65);"Zcor=";Zc
16340 Count=Kount
16350 PRINTER IS 16
16390 L$=L$&Comma$&F$
16400 PRINT PAGE;LIN(20)
16410   PRINT "FILES LINKED=";L$
16420 PRINT "TOTAL SAMPLES=";Count
16430 IF Count=400 THEN 16490
16440 INPUT "ANOTHER FILE? ",A$
16450 IF A$="N" THEN End
16451 Comma$=CHR$(44)
16460 INPUT "FILE NAME? ",F$
16480 GOTO Assign
16490 End:PRINTER IS 0
16495 PRINT "FILES LINKED=";L$
16500 PRINT "TOTAL SAMPLES=";Count
16510 PRINT
16520 INPUT "STORE DATA SET? ",S$
16530 IF S$="Y" THEN Store
16540 SUBEXIT
16550 Store:CALL Store(W(*),X(*),Y(*),Z(*),R1(*),R2(*),Time$(*),Count,F$)
16560 SUBEND
```

```
16570 SUB Stat_mat(W(*),X(*),Y(*),Z(*),S(*),Co(*),O(*),N)
16580 OPTION BASE 1
16590 DEFAULT ON
16600 DIM A(400),B(400),C(400),D(400),V(4)
16610 ! OFF-SET
16620 O(1)=W(1)
16630 O(2)=X(1)
16640 O(3)=Y(1)
16650 O(4)=Z(1)
16660 ! SUBTRACT OFF-SET
16670 FOR I=1 TO N
16680 A(I)=W(I)-O(1)
16690 B(I)=X(I)-O(2)
16700 C(I)=Y(I)-O(3)
16710 D(I)=Z(I)-O(4)
16720 NEXT I
16730 ! SUMS
16740 S(1,1)=SUM(A)
16750 S(1,2)=SUM(B)
16760 S(1,3)=SUM(C)
16770 S(1,4)=SUM(D)
16780 ! SUMS OF SQUARES
16790 S(2,1)=DOT(A,A)
16800 S(2,2)=DOT(B,B)
16810 S(2,3)=DOT(C,C)
16820 S(2,4)=DOT(D,D)
16830 ! MEAN AND STAN DEV
16840 FOR I=1 TO 4
16850 S(3,I)=S(1,I)/N
16860 V(I)=S(2,I)/N-S(3,I)^2
16870 S(4,I)=SQR(V(I))
16880 NEXT I
16890 ! SUM OF CROSS PRODUCTS
16900 Co(1,1)=DOT(A,B)
16910 Co(1,2)=DOT(A,C)
16920 Co(1,3)=DOT(A,D)
16930 Co(1,4)=DOT(B,C)
16940 Co(1,5)=DOT(B,D)
16950 Co(1,6)=DOT(C,D)
16960 ! COVARIANCE
16970 Co(2,1)=Co(1,1)/N-S(3,1)*S(3,2)
16980 Co(2,2)=Co(1,2)/N-S(3,1)*S(3,3)
16990 Co(2,3)=Co(1,3)/N-S(3,1)*S(3,4)
17000 Co(2,4)=Co(1,4)/N-S(3,2)*S(3,3)
17010 Co(2,5)=Co(1,5)/N-S(3,2)*S(3,4)
17020 Co(2,6)=Co(1,6)/N-S(3,3)*S(3,4)
17030 ! CORRELATION COEFFICIENT
17040 Co(3,1)=Co(2,1)/(S(4,1)*S(4,2))
17050 Co(3,2)=Co(2,2)/(S(4,1)*S(4,3))
17060 Co(3,3)=Co(2,3)/(S(4,1)*S(4,4))
17070 Co(3,4)=Co(2,4)/(S(4,2)*S(4,3))
17080 Co(3,5)=Co(2,5)/(S(4,2)*S(4,4))
17090 Co(3,6)=Co(2,6)/(S(4,3)*S(4,4))
17100 SUBEND
17120 !
```

```
17130 SUB Store(W(*),X(*),Y(*),Z(*),R1(*),R2(*),U$(*),N,F$)
17140 OPTION BASE 1
17150 DIM A(100),B(100),C(100),D(100),T$(100),E(100),F(100)
17160 IF N<=100 THEN Set=1
17170 IF (N>100) AND (N<=200) THEN Set=2
17180 IF (N>200) AND (N<=300) THEN Set=3
17190 IF (N>300) AND (N<=400) THEN Set=4
17200 INPUT " FILE NAME? ",F$
17210 PRINT "FILE=";F$,"SETS=";Set,"N=";N
17220 CREATE F$,Set,7000
17230 ASSIGN #1 TO F$
17240 PRINT #1;Set
17250 PRINT #1;N
17260 J=0
17270 Load:!
17280 FOR I=1 TO 100
17290 IF I+J*100>N THEN Zero
17300 A(I)=W(I+J*100)
17310 B(I)=X(I+J*100)
17320 C(I)=Y(I+J*100)
17330 D(I)=Z(I+J*100)
17340 E(I)=R1(I+J*100)
17350 F(I)=R2(I+J*100)
17360 T$(I)=U$(I+J*100)
17370 GOTO Next
17380 Zero:!
17390 A(I)=0
17400 B(I)=0
17410 C(I)=0
17420 D(I)=0
17430 E(I)=0
17440 F(I)=0
17450 T$(I)=""
17460 Next:NEXT I
17470 ! ON END #1 GOTO 16770
17480 ! PAUSE
17490 PRINT #1;A(*),B(*),C(*),D(*),E(*),F(*),T$(*)
17500 IF J+1=Set THEN SUBEXIT
17510 J=J+1
17520 GOTO Load
17530 SUBEND
17550 !
```

```
17560 SUB Delete_blok(W(*),X(*),Y(*),Z(*),R1(*),R2(*),T$(*),N,F$)
17570 OPTION BASE 1
17580 DIM We(400),Xe(400),Ye(400),Ze(400),Te$(400),R1e(400),R2e(400)
17590 PRINTER IS 0
17600 Sample:INPUT "FIRST SAMPLE IN BLOCK TO BE DELETED",N1
17610 INPUT "LAST SAMPLE IN BLOCK TO BE DELETED",N2
17620 J=0
17630 FOR I=1 TO N
17640 Sampl:IF (I>=N1) AND (I<=N2) THEN Next
17650 J=J+1
17660 DISP I,J
17670 We(J)=W(I)
17680 Xe(J)=X(I)
17690 Ye(J)=Y(I)
17700 Ze(J)=Z(I)
17710 R1e(J)=R1(I)
17720 R2e(J)=R2(I)
17730 Te$(J)=T$(I)
17740 Next: NEXT I
17750 N=J
17760 MAT W=We
17770 MAT X=Xe
17780 MAT Y=Ye
17790 MAT Z=Ze
17800 MAT R1=R1e
17810 MAT R2=R2e
17820 FOR I=1 TO N
17830 T$(I)=Te$(I)
17840 NEXT I
17850   PRINT "SAMPLES";N1;"THRU";N2;"DELETED,";N;"SAMPLES REMAIN"
17860 INPUT "ANOTHER BLOCK TO DELETE?",A$
17870 IF A$="Y" THEN Sample
17950 End:SUBEND
17970 !
```

```
17980 SUB Delete_td(W(*),X(*),Y(*),Z(*),R1(*),R2(*),T$(*),N,F$)
17990 OPTION BASE 1
18000 DIM We(400),Xe(400),Ye(400),Ze(400),Te$(400),R1e(400),R2e(400)
18001 Sec$="WXYZ"
18010 PRINTER IS 0
18020 Sample:Ni=N
18030 INPUT "SET CLIP LIMITS FOR WHICH SECONDARY? W1, X2, Y3, Z4 ",
Sec
18031 IF (Sec<1) OR (Sec>4) THEN Sample
18032 PRINT RPT$("-",80)
18037 PRINT "SETTING CLIP LIMITS FOR ";Sec$[Sec;1]
18040 INPUT "CLIP LIMITS ?; MIN,MAX",L1,U1
18041 PRINT "UPPER=";U1;" LOWER=";L1
18060 J=0
18070 FOR I=1 TO N
18080 Sampl:ON Sec GOTO Tw,Tx,Ty,Tz
18081 Tw:IF (W(I)<L1) OR (W(I)>U1) THEN Next
18082 GOTO Keep
18090 Tx:IF (X(I)<L1) OR (X(I)>U1) THEN Next
18091 GOTO Keep
18100 Ty:IF (Y(I)<L1) OR (Y(I)>U1) THEN Next
18101 GOTO Keep
18102 Tz:IF (Z(I)<L1) OR (Z(I)>U1) THEN Next
18110 Keep:J=J+1
18120 DISP I,J
18130 We(J)=W(I)
18140 Xe(J)=X(I)
18150 Ye(J)=Y(I)
18160 Ze(J)=Z(I)
18170 R1e(J)=R1(I)
18180 R2e(J)=R2(I)
18190 Te$(J)=T$(I)
18200 Next: NEXT I
18210 N=J
18220 MAT W=We
18230 MAT X=Xe
18240 MAT Y=Ye
18250 MAT Z=Ze
18260 MAT R1=R1e
18270 MAT R2=R2e
18280 FOR I=1 TO N
18290 T$(I)=Te$(I)
18300 NEXT I
18310 PRINT Ni-N;"SAMPLES DELETED,";N;"SAMPLES REMAIN"
18311 INPUT "SET ADDITIONAL CLIP LIMITS?",Q$
18312 IF UPC$(Q$[1,1])="Y" THEN Sample
18390 End:SUBEND
```

```
18400 SUB Track(Stat(*),O_set(*),Cov(*),R(*),Sample,F$,Wt,Wf,Angle,N)
18410 DEG
18420 OPTION BASE 1
18430 DEFAULT ON
18440 FIXED 2
18450 PRINT "LORAN-C POSITION ANALYSIS: ";F$
18451 Angle=Angle-180
18452 IF Angle<0 THEN Angle=Angle+360
18460   PRINT TAB(5);"FROM WP";Wf;" TO WP";Wt;",TRACK= ";Angle;" DEGREES"
18470 PRINT "REFERENCE WAYPOINT=";Wt
18480   Slope=R(1,1)
18490   IF R(3,1)=2 THEN Slope=1/Slope
18500   A=ATN(1/Slope)
18510   IF Slope<0 THEN A=A+180
18520   IF A<0 THEN A=A+360
18530   PRINT TAB(5);"RMS TRACKLINE=";A;" DEGREES"
18540   PRINT TAB(5);"AVE CROSS TRACK DISTANCE=";(O_set(3)+Stat(3,3))*1000;
" METERS"
18550   PRINT TAB(5);"STD DEV CROSS TRACK DISTANCE=";Stat(4,3)*1000;" METER
S"
18560 FIXED 3
18570 PRINT TAB(5);"AVE X POS=";O_set(1)+Stat(3,1);"KM";TAB(40);"Y POS=";O_
set(2)+Stat(3,2);"KM"
18580 PRINT TAB(5);"STANDARD DEV X=";Stat(4,1)*1000;" METERS";TAB(40);"Y=";
Stat(4,2)*1000;"METERS"
18590 SUBEND
```

```
18600  SUB Triangle(Sa,Sb,Sc,A,B,C)
18610  INTEGER E
18620  DEG
18630  S=(Sa+Sb+Sc)/2
18640  ON ERROR GOTO 18660
18650  GOTO 18680
18660 PRINT "ERROR 25";A;S;Sa;Sb;Sc
18670 SUBEXIT
18680  A=ACS(2*S*(S-Sa)/(Sb*Sc)-1)
18690  D=Sa/SIN(A)
18700  B=ASN(Sb/D)
18710  C=ASN(Sc/D)
18720  OFF ERROR
18730  E=A+B+C
18740  IF (A<90) AND (E<>180) THEN Alt
18750  SUBEXIT
18760 Alt:  IF Sb<Sc THEN Alt2
18770  B=180-B
18780  E=A+B+C
18790  SUBEXIT
18800 Alt2:  C=180-C
18810  E=A+B+C
18820 SUBEND
```

```
18830 SUB Position(Zo(*),Alpha,Sc,Sign,B,X,Y)
18840 DEG
18850 OPTION BASE 1
18860 Theta=Alpha+Sign*B
18870 X=Zo(1)+Sc*SIN(Theta)
18880 Y=Zo(2)+Sc*COS(Theta)
18890 SUBEND
```

```
18900 SUB Dif(Stat(*),O_set(*),N,F$,Re,Ree)
18910 DEG
18920 OPTION BASE 1
18930 DEFAULT ON
18940 PRINT "ANALYSIS OF LORAN-C VS MINI-RANGER DATA; FILE=";F$
18950 PRINT "-----------------------------------------------------"
18960 PRINT TAB(5);"AVE ERROR IN X-DIRECTION=";(O_set(3)+Stat(3,3))*1000;"M
ETERS"
18970 Rms=(Stat(2,3)/N+2*O_set(3)*(O_set(3)+Stat(3,3))+O_set(3)^2)^.5*1000
18980 PRINT TAB(5);"RMS ERROR IN X-DIRECTION=";Rms;"METERS"
18990 PRINT TAB(5);"AVE ERROR IN Y-DIRECTION=";(O_set(4)+Stat(3,4))*1000;"M
ETERS"
19000 Rms=(Stat(2,4)/N+2*O_set(4)*(O_set(4)+Stat(3,4))+O_set(4)^2)^.5*1000
19010 PRINT TAB(5);"RMS ERROR IN Y-DIRECTION=";Rms;"METERS"
19020 PRINT
19030 PRINT TAB(5);"AVE CROSS TRACK ERROR=";(O_set(2)+Stat(3,2))*1000;"METE
RS"
19040 Rms=(Stat(2,2)/N+2*O_set(2)*(O_set(2)+Stat(3,2))+O_set(2)^2)^.5*1000
19050 PRINT TAB(5);"RMS CROSS TRACK ERROR=";Rms;"METERS"
19060 PRINT TAB(5);"AVE ALONG TRACK ERROR=";(O_set(1)+Stat(3,1))*1000;"METE
RS"
19070 Rms=(Stat(2,1)/N+2*O_set(1)*(O_set(1)+Stat(3,1))+O_set(1)^2)^.5*1000
19080 PRINT TAB(5);"RMS ALONG TRACK ERROR=";Rms;"METERS"
19090 PRINT
19100 PRINT TAB(5);"RMS RADIAL ERROR(XY)=";Re;"METERS"
19110 PRINT TAB(5);"RMS RADIAL ERROR(CT/AT)=";Ree;"METERS"
19120 SUBEND
```

```
19130 SUB Plot_err(Ex(*),Ey(*),N,At(*),R,P)
19140 PLOTTER IS 13,"GRAPHICS"
19150 GRAPHICS
19160 LOCATE 0,100,55,100
19170 CALL Hi_lo(At(*),N,Xmax,Xmin)
19180 Xmin=MIN(0,Xmin)
19190 Xmax=MAX(Xmax,R)
19200 SCALE Xmin,Xmax,-.10,.10
19210 AXES .10,.01,0,0,10,10
19220 FOR I=1 TO N
19230 LORG 5
19240   MOVE At(I),Ex(I)
19250     LABEL USING "#,K";"X"
19260 ! MOVE I,Ey(I)
19270 NEXT I
19280 MOVE R,0
19290 LABEL USING "#,K";"I"
19291 MOVE .5,.10
19292 LORG 3
19293 LABEL USING "#,K";"100 M"
19294 LORG 1
19295 MOVE .5,-.10
19296 LABEL USING "#,K";"-100 M"
19297 LORG 5
19300 MOVE R/2,-.05
19310 IF P=2 THEN 19340
19320 LABEL USING "#,K";"X-ERROR VS ALONG TRACK DISTANCE (KM)"
19330 GOTO 19350
19340 LABEL USING "#,K";"At-ERROR VS ALONG TRACK DISTANCE (KM)"
19350 LOCATE 0,100,0,45
19360 SCALE Xmin,Xmax,-.10,.10
19370 AXES .10,.01,0,0,10,10
19371 LORG 5
19380 FOR I=1 TO N
19390   MOVE At(I),Ey(I)
19400     LABEL USING "#,K";"Y"
19410  NEXT I
19420 MOVE R,0
19430 LABEL USING "#,K";"I"
19431 MOVE .5,.10
19432 LORG 3
19433 LABEL USING "#,K";"100 ns"
19434 LORG 1
19435 MOVE .5,-.10
19436 LABEL USING "#,K";"-100 ns"
19437 LORG 5
19440 MOVE R/2,-.05
19450 IF P=2 THEN 19480
19460 LABEL USING "#,K";"Y-ERROR VS ALONG TRACK DISTANCE (KM)"
19470 SUBEXIT
19480 LABEL USING "#,K";"Ct-ERROR VS ALONG TRACK DISTANCE (KM)"
19490 SUBEND
```

```
19500 SUB X_errplot(T,Ax(*),Ay(*),Bx(*),By(*),Minx,Maxx,Miny,Maxy,N)
19510 DIM L$[80]
19520 PLOTTER IS 13,"GRAPHICS"
19530 GRAPHICS
19540 LOCATE 0,100,10,100
19550 FRAME
19560  SHOW Minx-.1,Maxx+.1,Miny-.1,Maxy+.1
19570 AXES .1,.1,Minx,Miny,10,10
19580 GRID 2,2,Minx,Miny,1,1
19590 LINE TYPE T
19600 MOVE Ax(1),Ay(1)
19610 LORG 5
19620 CSIZE 1
19630 FOR I=1 TO N
19640 MOVE Ax(I),Ay(I)
19650 DRAW Bx(I),By(I)
19660 LABEL USING "#,K";"O"
19670 NEXT I
19680 Label:!
19690 LOCATE 0,100,0,10
19700 SCALE 0,100,0,10
19710 CSIZE 3.0
19720 L$="ERROR PLOT: MINI-RANGER FIX(.) LORAN-C FIX(o)"
19730 MOVE 50,5
19740 LABEL L$
19750 PAUSE
19760 INPUT "HARD COPY? Y OR N",H$
19770 IF H$="Y" THEN DUMP GRAPHICS
19780 EXIT GRAPHICS
19790 SUBEND
```

```
19800 SUB Warp(X(*),Y(*),Z(*),Zx(*),Zy(*),Zo(*),To(*),Zxmit(*),V,N,At(*),R,
Wt,Wf,Conf,F$)
19810 ! FUNCTION CALCULATES DIFFERENCE BETWEEN MEASURED TDs AND TDs
19820 ! PROJECTED FROM A NEARBY WAYPOINT USING HYPERBOLIC GEOMETRY
19830 ! INPUTS:
19840 !    X(*),Y(*),Z(*)- MEASURED TD ARRAYS( NOT NECESSARLY TDX,TDY,TDZ)
19850 !    Zx(*),Zy(*)- POSITION DATA CALCULATED FRON MINI-RANGER DATA
19860 !    Zo(*)- WAYPOINT POSITION
19870 !    To(*)- WAYPOINT TDs
19880 !    Zxmit(*)- TRANSMITTER XY POSITIONS
19890 !    At(*)- ALONG TRACK DISTANCE ARRAY
19900 !    Wt,Wf- WAYPOINT TO AND WAYPOINT FROM
19910 !    F$-DATA FILE NAME
19920 !    R- RANGE BETWEEN WAYPOINTS
19930 !    V- VELOCITY OF PROPOGATION
19940 !    N- NUMBER OF SAMPLES
19950 !    Conf- CHAIN CONFIGURATION; 1=XYZ, 2=WXY, 3=WXZ, 4=WYZ
19960 ! CALCULATED:
19970 !    Xe(*),Ye(*),Ze(*)-TD ERROR ARRAYS
19980 !    Stat(*),O_set(*),Cov(*)- STATISTICS OF TD ERRORs
19990 OPTION BASE 1
20000 DEFAULT ON
20010 DIM R(4),T1(3),T2(3),Xe(400),Ye(400),Ze(400),Stat(4,4),O_set(4),Cov(3
,6)
20020 DIM L$[80]
20030 MAT Xe=(0)
20040 MAT Ye=(0)
20050 MAT Ze=(0)
20060 ! CALCULATE h(Zo)=(Range_secondary-Range_master)/Velocity_propogation
20070 FOR I=1 TO 4
20080 R(I)=SQR((Zo(1)-Zxmit(I,1))^2+(Zo(2)-Zxmit(I,2))^2)
20090 NEXT I
20100 FOR I=1 TO 3
20110 T1(I)=(R(I)-R(4))/V
20120 NEXT I
20130 ! CALCULATE h(Zi); Error=(TDi-TDo)-(h(Zi)-h(Zo))
20140 FOR I=1 TO N
20150 ! CALCULATE h(Zi)
20160 FOR J=1 TO 4
20170 R(J)=SQR((Zx(I)-Zxmit(J,1))^2+(Zy(I)-Zxmit(J,2))^2)
20180 NEXT J
20190 FOR J=1 TO 3
20200 T2(J)=(R(J)-R(4))/V
20210 NEXT J
20220 ! CALCULATE Error=(TDi-TDo)-(h(Zi)-h(Zo))
20230   Xe(I)=X(I)-To(1)-(T2(1)-T1(1))
20240   Ye(I)=Y(I)-To(2)-(T2(2)-T1(2))
20250   Ze(I)=Z(I)-To(3)-(T2(3)-T1(3))
20260 NEXT I
20270 Smooth:!
20280 FOR I=2 TO N-1
20290 Xe(I)=.7*Xe(I-1)+.3*Xe(I)
20300 Ye(I)=.7*Ye(I-1)+.3*Ye(I)
20310 Ze(I)=.7*Ze(I-1)+.3*Ze(I)
20320 NEXT I
```

```
20330 ! CALCULATE TD ERROR STATISTICS
20340 CALL Stat_mat(At(*),Xe(*),Ye(*),Ze(*),Stat(*),Cov(*),O_set(*),N)
20350 ! PRINT RESULTS
20360 IF Conf=1 THEN L$="TDX-ERROR       TDY-ERROR       TDZ-ERROR"
20370 IF Conf=2 THEN L$="TDW-ERROR       TDX-ERROR       TDY-ERROR"
20380 IF Conf=3 THEN L$="TDW-ERROR       TDX-ERROR       TDZ-ERROR"
20390 IF Conf=4 THEN L$="TDW-ERROR       TDY-ERROR       TDZ-ERROR"
20400 FIXED 0
20410 PRINT "FILE=";F$;"      REFERENCED TO WAYPOINT";Wt;":"
20420 PRINT TAB(10);L$
20430 FIXED 3
20440 PRINT "AVE";TAB(10);Stat(3,2)+O_set(2);TAB(25);Stat(3,3)+O_set(3);TAB
(40);Stat(3,4)+O_set(4)
20450 PRINT "STD DEV";TAB(10);Stat(4,2);TAB(25);Stat(4,3);TAB(40);Stat(4,4)
20460 ! PLOT DATA
20470   CALL T_errplot(Xe(*),Ye(*),Ze(*),At(*),R,N,Conf,Wt,Wf,F$)
20480 SUBEND
20490  !
```

```
20500 SUB T_errplot(A(*),B(*),C(*),At(*),R,N,Conf,Wt,Wf,F$)
20510  DEG
20520 ! PLOT TD ERROR DATA VS ALONG TRACK DISTANCE
20530 !     CALCULATE MIN AND MAX OF ERROR ARRAYS
20540 CALL Hi_lo(A(*),N,Maxa,Mina)
20550 CALL Hi_lo(B(*),N,Maxb,Minb)
20560 CALL Hi_lo(C(*),N,Maxc,Minc)
20570 !     CALCULATE MIN AND MAX OF ALONG TRACK DATA
20580 CALL Hi_lo(At(*),N,Maxat,Minat)
20590 !     GENERATE Y-AXIS LABELS
20600 IF Conf=1 THEN L1$="TDX"
20610 IF Conf<>1 THEN L1$="TDW"
20620 IF (Conf=2) OR (Conf=3) THEN L2$="TDX"
20630 IF (Conf=1) OR (Conf=4) THEN L2$="TDY"
20640 IF Conf<>2 THEN L3$="TDZ"
20650 !     PLOT A(*) DATA ON TOP QUARTER OF SCREEN
20660 PLOTTER IS 13,"GRAPHICS"
20670 GRAPHICS
20680 LORG 5
20690 LOCATE 10,120,75,100
20700 Miny=MIN(-.1,Mina)
20710 Maxy=MAX(.1,Maxa)
20720 Minx=MIN(0,Minat)
20730 Maxx=MAX(R,Maxat)
20740 SCALE Minx,Maxx,Miny,Maxy
20750 AXES .1,.01,0,0,10,10
20760 MOVE At(1),A(1)
20770 FOR I=1 TO N
20780 DRAW At(I),A(I)
20790 NEXT I
20800 MOVE R,0
20810 LABEL "I"
20820 MOVE -(Maxat-Minat)/30,0
20830 LDIR 90
20840 LABEL L1$
20850 LDIR 0
20860 !      PLOT B(*) DATA
20870 LOCATE 10,120,48,73
20880 Miny=MIN(-.1,Minb)
20890 Maxy=MAX(.1,Maxb)
20900 SCALE Minx,Maxx,Miny,Maxy
20910 AXES .1,.01,0,0,10,10
20920 MOVE At(1),B(1)
20930 FOR I=1 TO N
20940 DRAW At(I),B(I)
20950 NEXT I
20960 MOVE R,0
20970 LABEL "I"
20980 MOVE -(Maxat-Minat)/30,0
20990 LDIR 90
21000 LABEL L2$
21010 LDIR 0
21020 !       PLOT C(*) DATA
21030 LOCATE 10,120,20,45
21040 Miny=MIN(-.1,Minc)
```

```
21050 Maxy=MAX(.1,Maxc)
21060 SCALE Minx,Maxx,Miny,Maxy
21070 AXES .1,.01,0,0,10,10
21080 MOVE At(1),C(1)
21090 FOR I=1 TO N
21100 DRAW At(I),C(I)
21110 NEXT I
21120 MOVE R,0
21130 LABEL "I"
21140 MOVE -(Maxat-Minat)/30,0
21150 LDIR 90
21160 LABEL L3$
21170 LDIR 0
21180 !         LABEL
21190 FIXED 0
21200 LOCATE 0,120,0,20
21210 SCALE 0,120,0,20
21220 MOVE 10,18
21230 LABEL Wt
21240 MOVE 120,18
21250 LABEL Wf
21260 MOVE 60,15
21270 LABEL "VERTICAL=100 NANOSEC/MAJOR DIV    HORIZONTAL=1 KM/MAJOR DIV"
21280 MOVE 60,12
21290 LABEL "SMOOTHED TD ERROR VS ALONG TRACK DISTANCE"
21300 W$=VAL$(Wt)
21310 MOVE 60,9
21320 LABEL "REFERENCED TO WAYPOINT "&W$
21330 MOVE 60,6
21340 LABEL "FILE="&F$
21350 !           HARD COPY
21360 PAUSE
21370 INPUT "HARD COPY? Y OR N",Hc$
21380 IF Hc$="Y" THEN DUMP GRAPHICS
21390 EXIT GRAPHICS
21400 SUBEND
```

```
21410 SUB Pseudo(Glat,Glon,R(*),Plat,Plon)
21420 OPTION BASE 1
21430 RAD
21440 DIM X(3),T(3)
21450 T(1)=COS(Glat)*COS(Glon)
21460 T(2)=COS(Glat)*SIN(Glon)
21470 T(3)=SIN(Glat)
21480 MAT X=R*T
21490 H=SQR(X(1)^2+X(2)^2)
21500 Plat=ATN(X(3)/H)
21510 Plon=ATN(X(2)/X(1))
21520 SUBEND
21530 !
```

```
21600 SUB Rotate(Glat,Glon,R(*))
21610 RAD
21620 OPTION BASE 1
21630 R(1,1)=COS(Glat)*COS(Glon)
21640 R(2,1)=-SIN(Glon)
21650 R(3,1)=-SIN(Glat)*COS(Glon)
21660 R(1,2)=COS(Glat)*SIN(Glon)
21670 R(2,2)=COS(Glon)
21680 R(3,2)=-SIN(Glat)*SIN(Glon)
21690 R(1,3)=SIN(Glat)
21700 R(2,3)=0
21710 R(3,3)=COS(Glat)
21720 SUBEND
21740 !
```

```
21750 SUB Cart(Plat,Plon,X,Y)
21760 OPTION BASE 1
21770 RAD
21780 R=6378.135
21790 Cos=COS(Plat)*COS(Plon)
21800 Sin=SQR(1-Cos^2)
21810 Theta=ATN(Sin/Cos)
21820 S=R*Theta
21830 Sin=SIN(Plat)/SIN(Theta)
21840 Cos=SIN(Plon)*COS(Plat)/SIN(Theta)
21850 X=S*Cos
21860 Y=S*Sin
21870 SUBEND
21890 !
```

```
21900 DEF FNGlat(Rlat,F)
21910 RAD
21920 RETURN ATN((1-F)^2*TAN(Rlat))
21930 FNEND
21950 !
```

```
21960 SUB Ref_file(R(*),R$(*),Rp$)
21970 ! PROGRAM TO STORE POSITION REFERENCE DATA
21980 OPTION BASE 1
21990 PRINTER IS 0
22010 INPUT "REFERENCE POSITION FILE NAME?",Rp$
22020 F2=1
22030 INPUT "IS THIS FILE PRESENTLY ON TAPE?",Ex$
22040 IF UPC$(Ex$)="N" THEN CREATE Rp$,4
22060 Cont:ASSIGN #1 TO Rp$
22070 IF UPC$(Ex$)="Y" THEN READ #1;R$(*),R(*)
22090 Opt:INPUT "MODIFY/CREATE1, LIST2, SAVE3, QUIT4",Opt
22100 IF (Opt<1) OR (Opt>4) THEN Opt
22110 ON Opt GOTO Input,List,File,End
22160 Input:FOR N=1 TO 3
22170 INPUT "LABEL?",R$(N)
22270 INPUT "X-POSITION?",R(N,1)
22280 INPUT "Y-POSITION?",R(N,2)
22290 PRINT N;R$(N);TAB(20);R(N,1);TAB(30);R(N,2)
22291 NEXT N
22300 GOTO Opt
22310 List: PRINT LIN(5);"POSITION REFERENCE FILE: ";Rp$
22311 FOR I=1 TO 3
22320 PRINT I;R$(I);TAB(25);R(I,1);TAB(35);R(I,2)
22330 NEXT I
22331 PRINT LIN(5)
22340 GOTO Opt
22350 File:ASSIGN #1 TO Rp$
22360 PRINT #1,1;R$(*),R(*)
22370 End:SUBEND
22371 !
```

```
22380 SUB Delete_rg(W(*),X(*),Y(*),Z(*),R1(*),R2(*),T$(*),N,F$)
22390 OPTION BASE 1
22400 DIM We(400),Xe(400),Ye(400),Ze(400),Te$(400),R1e(400),R2e(400)
22410 PRINTER IS 0
22420 J=0
22430 FOR I=1 TO N
22440 IF (I=1) OR (I=N) THEN 22470
22450 Sampl:  IF ABS(R1(I)-(R1(I+1)+R1(I-1))/2)>100 THEN Next
22460   IF ABS(R2(I)-(R2(I+1)+R2(I-1))/2)>100 THEN Next
22470 J=J+1
22480 DISP I,J
22490 We(J)=W(I)
22500 Xe(J)=X(I)
22510 Ye(J)=Y(I)
22520 Ze(J)=Z(I)
22530 R1e(J)=R1(I)
22540 R2e(J)=R2(I)
22550 Te$(J)=T$(I)
22560 GOTO 22580
22570 Next:PRINT "SAMPLE DELETED=";I
22580 NEXT I
22590 Delete=N-J
22600 N=J
22610 MAT W=We
22620 MAT X=Xe
22630 MAT Y=Ye
22640 MAT Z=Ze
22650 MAT R1=R1e
22660 MAT R2=R2e
22670 FOR I=1 TO N
22680 T$(I)=Te$(I)
22690 NEXT I
22700   PRINT "SAMPLES DELETED=";Delete;" SAMPLES REMAINING=";N
22720 End:SUBEND
22740 !
```

```
22750 SUB Cart_coord(X(*),Lat,Lon,Z(*))!
22760 OPTION BASE 1
22770 ! CALCULATE XY COORDINATES OF TRANSMITTERS(Z(*)) WITH RESPECT
22780 ! TO LOCAL GRID ORIGION(Lat,Lon)
22790 ! INPUT: TRANSMITTER GEODETIC POSITIONS;X(*)
22800 !         LAT AND LON OF LOCAL GRID ORIGION; Lat,Lon
22810 ! OUTPUT: TRANSMITTER XY POSITIONS; Z(*)
22820 ! SEE APL TECH NOTE
22830 DIM G(4,2),P(4,2),Rotate(3,3)
22840 F=.00335278
22850 Rlat=Lat*PI/180
22860 Rlat=FNGlat(Rlat,F)
22870 Rlon=Lon*PI/180
22880 CALL Rotate(Rlat,Rlon,Rotate(*))
22890 FOR I=1 TO 4
22900 G(I,1)=X(I,1)*PI/180
22910 G(I,1)=FNGlat(G(I,1),F)
22920 G(I,2)=X(I,2)*PI/180
22930 CALL Pseudo(G(I,1),G(I,2),Rotate(*),P(I,1),P(I,2))
22940 CALL Cart(P(I,1),P(I,2),Z(I,1),Z(I,2))
22950 NEXT I
22960 SUBEND
22961 !
```

```
23030 SUB Wp3(W(*),C,W,T(*),Z(*),L$)
23040 OPTION BASE 1
23050 ! SELECT THE 3 TDs TO BE USED AS THE WAYPOINT(T(*)) FOR THE
23060 ! 3-TD FEHG SOLUTION BASED ON THE CHAIN CONFIGURATION(C)
23070 ! INPUTS:
23080 !        W(*); WAYPOINT TABLE
23090 !        C; CHAIN CONFIGURATION, 1=XYZ, 2=WXY, 3=WXZ, 4=WYZ
23100 !        W; WAYPOINT NUMBER
23110 ! OUTPUTS:
23120 !        T(*); WAYPOINT TDs
23130 !        Z(*); WAYPOINT XY POSITION
23140 !        L$; LABEL
23150     Z(1,1)=W(W,5)
23160     Z(2,1)=W(W,6)
23170 ON C GOTO Xyz,Wxy,Wxz,Wyz
23180 Xyz:!
23190     T(1,1)=W(W,2)
23200     T(2,1)=W(W,3)
23210     T(3,1)=W(W,4)
23220     L$="XYZ"
23230     SUBEXIT
23240 Wxy:     !
23250     T(1,1)=W(W,1)
23260     T(2,1)=W(W,2)
23270     T(3,1)=W(W,3)
23280     L$="WXY"
23290     SUBEXIT
23300 Wxz:!
23310     T(1,1)=W(W,1)
23320     T(2,1)=W(W,2)
23330     T(3,1)=W(W,4)
23340     L$="WXZ"
23350     SUBEXIT
23360 Wyz:     !
23370     T(1,1)=W(W,1)
23380     T(2,1)=W(W,3)
23390     T(3,1)=W(W,4)
23400     L$="WYZ"
23410 SUBEND
23430 !
```

```
23440 SUB Td3(W(*),X(*),Y(*),Z(*),I,C,T(*))
23450 OPTION BASE 1
23460 ! SELECTS THE PROPER TD SAMPLES FOR CALCULATING XY POSITION
23470 ! BASED ON CHAIN CONFIGURATION AND 3-TD FEHG SOLUTION
23480 ! INPUTS:
23490 !       W(*),X(*),Y(*),Z(*); TD DATA ARRAYS
23500 !       I; SAMPLE NUMBER
23510 !       C; CHAIN CONFIGURATION 1=XYZ, 2=WXY, 3=WXZ, 4=WYZ
23520 ! OUTPUT:
23530 !       T(*); TD SAMPLE
23540 ON C GOTO Xyz,Wxy,Wxz,Wyz
23550 Xyz:!
23560     T(1,1)=X(I)
23570     T(2,1)=Y(I)
23580     T(3,1)=Z(I)
23590     SUBEXIT
23600 Wxy:    !
23610    T(1,1)=W(I)
23620    T(2,1)=X(I)
23630    T(3,1)=Y(I)
23640    SUBEXIT
23650 Wyz:   !
23660    T(1,1)=W(I)
23670    T(2,1)=Y(I)
23680    T(3,1)=Z(I)
23690    SUBEXIT
23700 Wxz:   !
23710    T(1,1)=W(I)
23720    T(2,1)=X(I)
23730    T(3,1)=Z(I)
23740 SUBEND
23760 !
```

```
23770 SUB Wp2(W(*),C,W,T(*),P,Z(*),G12(*),G23(*),G13(*),G(*),L$,Zx(*),Zxt(*
))
23780 OPTION BASE 1
23790 ! SELECT WAYPOINT TDs,POSITION,TRANSMITTER POSITIONS AND G-MATRIX FOR
 2-TD FEHG SOLUTION
23800 ! FEHG SOLUTION BASED ON CHAIN CONFIGURATION AND DESIRED TD PAIR
23810 ! INPUTS:
23820 !       W(*); WAYPOINT TABLE
23830 !       C; CHAIN CONFIGURATION 1=XYZ, 2=WXY, 3=WXZ, 4=WYZ
23840 !       P; TD PAIR
23850 !       G12(*),G23(*),G13(*); 2-TD G-MATRICES
23860 !       Zx(*); TRANSMITTER POSITIONS
23870 ! OUTPUTS:
23880 !       T(*); WAYPOINT TDs
23890 !       G(*); G-MATRIX
23900 !       Z(*); WAYPOINT POSITION
23910 !       Zxt(*); TRANSMITTER POSITIONS FOR 2-TD SOLUTION
23920 !       L$; LABEL
23930 Z(1,1)=W(W,5)
23940 Z(2,1)=W(W,6)
23950 Zxt(3,1)=Zx(4,1)
23960 Zxt(3,2)=Zx(4,2)
23970 ON C GOTO Xyz,Wxy,Wxz,Wyz
23980 Xyz:!
23990 INPUT "INPUT TD PAIR; XY1, XZ2, YZ3",P
24000 ON P GOTO Xy,Xz,Yz
24010 Wxy:!
24020 INPUT "INPUT TD PAIR; WX1, WY2, XY3",P
24030 ON P GOTO Wx,Wy,Xy
24040 Wxz:!
24050 INPUT "INPUT TD PAIR; WX1, WZ2, XZ3",P
24060 ON P GOTO Wx,Wz,Xz
24070 Wyz:!
24080 INPUT "INPUT TD PAIR; WY1, WZ2, YZ3",P
24090 ON P GOTO Wy,Wz,Yz
24100 Wx:!
24110   T(1,1)=W(W,1)
24120   T(2,1)=W(W,2)
24130   L$="WX"
24140   GOTO Xmit_sel
24150 Wy:  !
24160   T(1,1)=W(W,1)
24170   T(2,1)=W(W,3)
24180   L$="WY"
24190   GOTO Xmit_sel
24200 Wz:  !
24210   T(1,1)=W(W,1)
24220   T(2,1)=W(W,4)
24230   L$="WZ"
24240   GOTO Xmit_sel
24250 Xy:  !
24260   T(1,1)=W(W,2)
24270   T(2,1)=W(W,3)
24280   L$="XY"
24290   GOTO Xmit_sel
```

```
24300 Xz:  !
24310   T(1,1)=W(W,2)
24320   T(2,1)=W(W,4)
24330   L$="XZ"
24340   GOTO Xmit_sel
24350 Yz:  !
24360   T(1,1)=W(W,3)
24370   T(2,1)=W(W,4)
24380   L$="YZ"
24390   GOTO Xmit_sel
24400 Xmit_sel:  !
24410 ON P GOTO X12,X13,X23
24420 X12:!
24430    MAT G=G12
24440    Zxt(1,1)=Zx(1,1)
24450    Zxt(1,2)=Zx(1,2)
24460    Zxt(2,1)=Zx(2,1)
24470    Zxt(2,2)=Zx(2,2)
24480    SUBEXIT
24490 X13:    !
24500    MAT G=G13
24510    Zxt(1,1)=Zx(1,1)
24520    Zxt(1,2)=Zx(1,2)
24530    Zxt(2,1)=Zx(3,1)
24540    Zxt(2,2)=Zx(3,2)
24550    SUBEXIT
24560 X23:    !
24570    MAT G=G23
24580    Zxt(1,1)=Zx(2,1)
24590    Zxt(1,2)=Zx(2,2)
24600    Zxt(2,1)=Zx(3,1)
24610    Zxt(2,2)=Zx(3,2)
24620   SUBEND
24640 !
```

```
24650 SUB Td2(W(*),X(*),Y(*),Z(*),I,C,P,T(*))
24660 ! SELECT THE PROPER TD SAMPLES FOR CALCULATING XY POSITION
24670 ! BASED ON CHAIN CONFIGURATION, TD PAIR, AND 2-TD FEHG SOLUTION
24680 ! INPUTS:
24690 !       W(*),X(*),Y(*),Z(*); TD DATA ARRAYS
24700 !       I; SAMPLE NUMBER
24710 !       C; CHAIN CONFIGURATION
24720 !       P; TD PAIR
24730 ! OUTPUT:
24740 !       T(*); TD SAMPLE
24750 OPTION BASE 1
24760 ON C GOTO Xyz,Wxy,Wxz,Wyz
24770 Xyz:!
24780 ON P GOTO Xy,Xz,Yz
24790 Wxy:!
24800 ON P GOTO Wx,Wy,Wz
24810 Wxz:!
24820 ON P GOTO Wx,Wz,Xz
24830 Wyz:!
24840 ON P GOTO Wy,Wz,Yz
24850 Wx:!
24860 T(1,1)=W(I)
24870 T(2,1)=X(I)
24880 SUBEXIT
24890 Wy:!
24900 T(1,1)=W(I)
24910 T(2,1)=Y(I)
24920 SUBEXIT
24930 Wz:!
24940 T(1,1)=W(I)
24950 T(2,1)=Z(I)
24960 SUBEXIT
24970 Xy:!
24980 T(1,1)=X(I)
24990 T(2,1)=Y(I)
25000 SUBEXIT
25010 Xz:!
25020 T(1,1)=X(I)
25030 T(2,1)=Z(I)
25040 SUBEXIT
25050 Yz:!
25060 T(1,1)=Y(I)
25070 T(2,1)=Z(I)
25080 SUBEND
25100 !
```

```
25110 SUB Wpf2(W(*),C,P,W,T(*))
25120 ! SELECTS PROPER TDs FROM WAYPOINT FILE FOR CALCULATION OF
25130 ! THE "FROM" WAYPOINT POSITION USING THE 2-TD FEHG SOLUTION
25140 ! INPUTS:
25150 !        W(*); WAYPOINT FILE
25160 !        C; CHAIN CONFIGURATION
25170 !        P; TD PAIR
25180 !        W; WAYPOINT FROM
25190 ! OUTPUT:
25200 !        T(*); 2-TD WAYPOINT FOR WAYPOINT FROM
25210 ON C GOTO Xyz,Wxy,Wxz,Wyz
25220 Xyz: ON P GOTO Xy,Xz,Yz
25230 Wxy: ON P GOTO Wx,Wy,Xy
25240 Wyz: ON P GOTO Wy,Wz,Yz
25250 Wx:!
25260 T(1,1)=W(W,1)
25270 T(2,1)=W(W,2)
25280 SUBEXIT
25290 Wy:!
25300 T(1,1)=W(W,1)
25310 T(2,1)=W(W,3)
25320 SUBEXIT
25330 Wz:!
25340 T(1,1)=W(W,1)
25350 T(2,1)=W(W,4)
25360 SUBEXIT
25370 Xy:!
25380 T(1,1)=W(W,2)
25390 T(2,1)=W(W,3)
25400 SUBEXIT
25410 Xz:!
25420 T(1,1)=W(W,2)
25430 T(2,1)=W(W,4)
25440 SUBEXIT
25450 Yz:!
25460 T(1,1)=W(W,3)
25470 T(2,1)=W(W,4)
25480 SUBEND
25500 !
```

```
25510 SUB Rb(X1,Y1,X2,Y2,B,R)
25520 DEG
25530 DEFAULT ON
25540 Dx=X2-X1
25550 Dy=Y2-Y1
25560 R=SQR(Dx^2+Dy^2)
25570 B=ATN(Dx/Dy)
25580 IF Dy<0 THEN B=B+180
25590 IF B<0 THEN B=B+360
25600 SUBEND
25601 !
```

```
26180 SUB Compar(Pos_x(*),Pos_y(*),Zx(*),Zy(*),At(*),Ct(*),Att(*),Ctt(*),F$
,N,R)
26190 DIM Ae(400),Ce(400),Xe(400),Ye(400),Stat(4,4),Cov(3,6),O_set(4)
26200     ! COMPARE MINI-RANGER AND LORAN-C POSITIONS
26210  Re=0
26220  Ree=0
26230  FOR I=1 TO N
26240  Xe(I)=Pos_x(I)-Zx(I)
26250  Ye(I)=Pos_y(I)-Zy(I)
26260  Ce(I)=Ct(I)-Ctt(I)
26270  Ae(I)=At(I)-Att(I)
26280  Re=Re+((Xe(I)*1000)^2+(Ye(I)*1000)^2)
26290  Ree=Ree+((Ce(I)*1000)^2+(Ae(I)*1000)^2)
26300  NEXT I
26310  Re=(Re/N)^.5
26320  Ree=(Ree/N)^.5
26330  CALL Stat_mat(Ae(*),Ce(*),Xe(*),Ye(*),Stat(*),Cov(*),O_set(*),N)
26340  FIXED 3
26350  PRINTER IS 0
26360  CALL Dif(Stat(*),O_set(*),N,F$,Re,Ree)
26370  CALL Plot_err(Xe(*),Ye(*),N,Att(*),R,1)
26380  PAUSE
26390  INPUT "HARD COPY?,Y OR N",H$
26400  IF H$="Y" THEN DUMP GRAPHICS
26410  CALL Plot_err(Ae(*),Ce(*),N,Att(*),R,2)
26420  PAUSE
26430  INPUT "HARD COPY?,Y OR N",H$
26440  IF H$="Y" THEN DUMP GRAPHICS
26450  EXIT GRAPHICS
26460  SUBEND
```

```
26470  SUB Reflect(W,Wpt(*),W(*),X(*),Y(*),Z(*),Zx(*),Zy(*),Conf,Zxmit(*),N
,V)
26480  OPTION BASE 1
26490  DIM T1(400),T2(400),T3(400),Bear(4),Range(4),T(3)
26500  !  REFLECTS TDs MEASURED NEAR A WAYPOINT TO THE WAYPOINT
26510  !  TDreflect=TDmeas-(h(Zmeas)-h(Zwaypoint))
26520  !    h(Z)=(Range_secondary-Range_master)/Velocity_propogation
26530  !            CALCULATE h(Zwaypoint)
26540  FOR I=1 TO 4
26550  CALL Rb(Wpt(W,5),Wpt(W,6),Zxmit(I,1),Zxmit(I,2),Bear(I),Range(I))
26560  NEXT I
26570  FOR I=1 TO 3
26580  T(I)=(Range(I)-Range(4))/V
26590  NEXT I
26600  !            CALCULATE h(Zmeas)-h(Zwaypoint)
26610  FOR I=1 TO N
26620  FOR J=1 TO 4
26630  CALL Rb(Zx(I),Zy(I),Zxmit(J,1),Zxmit(J,2),Bear(J),Range(J))
26640  NEXT J
26650  T1(I)=(Range(1)-Range(4))/V-T(1)
26660  T2(I)=(Range(2)-Range(4))/V-T(2)
26670  T3(I)=(Range(3)-Range(4))/V-T(3)
26680  NEXT I
26690  !            CALCULATE TDmeas-(h(Zmeas)-h(Zwaypoint))
26700  ON Conf GOTO Xyz,Wxy,Wxz,Wyz
26710 Xyz: !
26720  MAT W=(0)
26730  MAT X=X-T1
26740  MAT Y=Y-T2
26750  MAT Z=Z-T3
26760  SUBEXIT
26770 Wxy: !
26780  MAT W=W-T1
26790  MAT X=X-T2
26800  MAT Y=Y-T3
26810  MAT Z=(0)
26820  SUBEXIT
26830 Wxz: !
26840  MAT W=W-T1
26850  MAT X=X-T2
26860  MAT Y=(0)
26870  MAT Z=Z-T3
26880  SUBEXIT
26890 Wyz: !
26900  MAT W=W-T1
26910  MAT X=(0)
26920  MAT Y=Y-T2
26930  MAT Z=Z-T3
26940  SUBEND
```