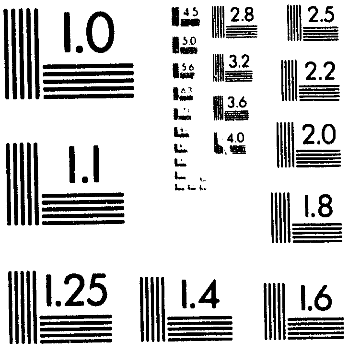

1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

LA-12644-M
Manual

UC-000
Issued: November 1993

TIPC User Manual

Thomas A. Kelley
Robert A. Roussel-Dupré
Eugene M. D. Symbalisty
William B. Clodius

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

TABLE OF CONTENTS

TIPC USER MANUAL

by

Thomas A. Kelley
Robert A. Roussel-Dupré
Eugene M. D. Symbalisty
William B. Clodius

ABSTRACT

The TransIonospheric Propagation Code (TIPC) computer program executes tasks related to the detection of VHF and UHF signals following propagation through the ionosphere. These tasks include: transionospheric propagation, signal detection, signal processing, delta time of arrival study, delta time of arrival uncertainty study, and signal reconstruction. The parameters needed to accomplish each task are defined and the process of using TIPC in each of these tasks is explained step-by-step. TIPC is also capable of saving these parameter values defined for the specific task. In addition, subroutine descriptions are identified.

1. Introduction to TransIonospheric Propagation Code

Description

The TransIonospheric Propagation Code (TIPC) is a computer program developed at Los Alamos National Lab(LANL) to perform certain tasks related to the detection of VHF signals following propagation through the ionosphere. The code is written in Fortran 77, runs interactively and is designed to be as machine-independent as possible. A menu format in which the user is prompted to supply appropriate parameters for a given task has been adopted for the input while the output is primarily in the form of graphics. The user has the option of selecting from eight basic tasks:

a) Propagate a specified input signal through a specified model ionosphere,
b) Detect the transionospheric signal with specified filters having various frequency response characteristics,
c) Apply signal processing techniques to recast the received signal in various forms and/or extract information about the ionosphere,
d) Determine the difference in arrival times of various frequency components (the central frequencies of specified narrowband receivers) of the received signals as a function of the total electron content (TEC) of the ionosphere,
e) Determine the effect of noise on the differences in arrival times of the various frequency components of the received signals,
f) Attempt to reconstruct the input signal based on a given set of filters and the signals received by them,
g) Save parameter values in a file, and
h) Restore parameter values from a file.

History

TIPC existed in various forms and in different pieces of code within SST-7 (Atmospheric Sciences group) at LANL. The pieces were amalgamated, and the code was heavily modified under contract to C-9 (Computer Applications group) at LANL. The user interface portion of TIPC was rewritten to take advantage of a library of routines called OINK (Output INput Kit). Additional changes were recently incorporated to allow operation on UNIX operating systems. The graphics portion of TIPC was rewritten so as to utilize either the graphics package provided by the National Center for Atmospheric Research or the DISSPLA graphics package distributed by Computer Associates or the Los Alamos graphics package entitled CGSHIGH. The SUN workstation and CRAY Unicos versions of the code utilize this latter graphics package.

Contents

The TIPC software package for the SUN comes in the following files:

Fortran files

tipc.f	The main program and most of the associated subroutines.
eshi.f	The graphics related routines.
oink.f	The user interface routines.
sun.f	Random number routines for the SUN.
fft.f	The Fast Fourier Transform routines.

menus.f	Menu display routines.
gets.f	Data retrieval related routines.
savwrt.f	Save and Write related routines.
tasks.f	Task driver routines.
initrd.f	Initialization and Read routines.
cray.f	CRAY Unicos specific routines.

Include files

tipc.inc	Definition of global parameters and variables in TIPC.
plot.inc	Definition of global parameters for the plot routines.
tty.inc	Definition of terminal types.
spcinc.inc	Definition of Spectrogram parameters and variables.

Make files

makesuncgs	Unix make file for the SUN with CGS graphics.
makedbug	Unix make file with debugging turned on and therefore optimization turned off.
tipc.x	Compile and load file for the CRAY Unicos environment.

Sample Parameter Files

TIPC.PRM	Default parameter file, chooses a signal processing task.
dtoa.prm	Delta time of arrival study parameter file.
delun.prm	Delta time of arrival uncertainty study parameter file.
dish.prm	Parameter file appropriate for using the tabulated signal in the file *dishnu.*

Sample Data File

dishnu	Time/Amplitude signal data from a Los Alamos transmitter.

Sample Batch File

test.com	Simple example of a command file for running TIPC in the batch mode.

Portability

TIPC is written in FORTRAN and currently runs in a VAX/VMS, CONVEX/UNIX, the SUN Openwindows environment, and the CRAY Unicos environment at Los Alamos. TIPC is designed to be as portable as possible. However, in some places, it does depend on software that may be site-specific. The graphics data interface is one such place. Here at Los Alamos, TIPC relies on our Common Graphics System (CGS) library. TIPC's dependence upon CGS has been isolated to the routines found in the file *eshi.f* and to the definition of associated global variables found in the file *PLOT.INC*. Another place TIPC is environment specific is in the generation of random numbers. The routine NEWSEED makes use of the way the VAX stores its system time to generate a seed value for the random number generator while the routine RANDU relies on a call to the VMS math library to generate uniform random numbers. A third place is in the user interface. The routine CLEAR relies on a VMS system call to wait for a second while a Tektronix screen is being cleared. These VMS dependent subroutines can be found in a separate file *VMS.FOR*. A fourth place is in the user interface. The OINK routines have been designed to work under three different systems: VAX/VMS , Los Alamos CRAY CTSS, and Bell Labs UNIX. An argument in the call to the initialization routine IOINIT specifies the system.

Construction

To build TIPC for the SUN environment, follow these steps:

1) Extract the files using the UNIX archive facility:

```
ar xv tiphi06.ar
```

2) Check to see that the temp directory is defined properly in the *makesuncgs* and *makedbug* files.

3) Make sure the following global parameters which are defined in TIPC.INC are appropriate for your environment.

EPSILON	small positive number to prevent division by zero
EXPMIN	largest argument for exp(-x)
MAXCOF	maximum number of coherence bandwidths
MAXDEL	maximum number of delta function for input signal
MAXRCVR	maximum number of receivers
MAXRFL	maximum number of reflections for supergaussian
MAXSNR	maximum number of SNR values
MAXTEC	maximum number of TEC values
NCYCLE	number of cycles for log plots
NIRF	number of random function values for scintillations
NPTS	number of values of sampled signal
TECONST	TEC constant

4) Make the executable, TIPC

```
make -f makesuncgs
```

or for debugging purposes,

```
make -f makedbug
```

The *makedbug* produces an executable entitled *tipdbug* which runs much slower than the standard TIPC executable, but is useful when debugging in the SUN environment.

2. Getting started

Running the Program

To run the program, enter the command **tipc**. The name of the code along with its current version number and date will be displayed. Finally you will be asked to choose a terminal type. Here is an example:

```
     T I P C         Version 6.0  March, 1993

Which type of terminal are you using?

:::::::::::::::::::::::::::::::::::
::                               ::
:: [1] No terminal - batch mode  ::
:: [2] DEC compatible            ::
:: [3] Xwindow                   ::
:: [4] <exit from program>       ::
::                               ::
:::::::::::::::::::::::::::::::::::
Your choice? <4>
```

Choosing a terminal type

Choose terminal type 1 if you want to run the program in batch mode. In this case, it is assumed that you have a set of parameter values stored away in a parameter file. TIPC will ask you which task you want to perform. Unless you reply with number 9 (exit the program), the code will then ask you for the name of a parameter file. After this, TIPC will perform the requested task without any further intervention on your part. When finished with the task, TIPC will return to the main menu, so to speak, and again ask you which task you want to perform. You can choose another task or you can exit the program by selecting task number 9. The appropriate responses can be built into a command file so that the running of TIPC can be completely automated. Warning: you must be very careful in constructing a command file to run TIPC in batch mode. TIPC is not very tolerant of errors in this regards. Here is an example of a command file designed for the SUN/UNIX system:

The contents of the *test.com* file are

```
1
4
8
TIPC.PRM
3
<return>
<return>
<return>
<return>
```

Choose terminal type 2 if you want to run interactively and you have a DEC compatible terminal. In this case, the graphics output will be saved in a graphics metafile for later display.

Choose terminal type 3 if you want to run interactively and you have a Xwindow terminal. In this case, the graphics output will be sent directly to your terminal and also saved in a graphics metafile for later display. This option also allows the user to redraw plots with different limits for the x-axis.

Choosing the number of colors

The user may choose the number of colors used for graphical output by selecting a number less than or equal to the number available on the local workstation. The number of colors is also equal to the number of contours drawn in the call to PLTXYZ.

Choosing a task to perform

After you enter your choice, the screen will be refreshed, and the main menu of options will be presented to you. It will look like this:

```
Choose Task to Perform

::::::::::::::::::::::::::::::::::::::
::                                  ::
:: [1] Transionospheric Propagation ::
:: [2] Signal Detection             ::
:: [3] Signal Processing            ::
:: [4] Delta Time of Arrival Study  ::
:: [5] DTOA uncertainty Study       ::
:: [6] Signal reconstruction        ::
:: [7] Save parameter values        ::
:: [8] Restore parameter values     ::
:: [9] <exit the program>           ::
::                                  ::
::::::::::::::::::::::::::::::::::::::
Your choice? <9>
```

The function performed by each of these tasks and the manipulation of the necessary parameters will be explained in the following sections.

After the necessary parameters have been defined, TIPC will attempt to perform the requested task. At various stages in the calculation, messages will be displayed for your benefit. These will be your clues first of all that TIPC is running and secondly what basic calculations are currently being performed. After the task is completed, TIPC will return to the main menu and again prompt you for a task to perform. The default task, which you may choose by simply pressing the return key, is to exit the program.

Hints on interacting with TIPC

While the program attempts to define the necessary parameters for performing the task you choose, it will present different menus to you and will no doubt ask you for specific information. Keep the following points in mind.

1. For all the replies that you will be asked to make, there are default values. These appear within the angle brackets < and >. These values will be used if you press only the return key. This makes it easy to accept parameters values that you do not want to change.

2. When you are prompted for a parameter value, you may ask for more explanation by entering a question mark and then pressing the return key. You may be told what the choices are, what the range of values are, or what the units are for the given parameter.

3. When you are prompted for a parameter value, you may go back to a previous question by entering the less than (<) and then pressing the return key.

4. You can abort the process of parameter definition and return back to the main menu of tasks at almost any time. This may be one of the menu options you can choose. If not, at any time you are being prompted for a parameter value, you can enter an exclamation point (!) and then press the return key.

You will be reminded of the three special keys (? < and !) at various times during the interactive phase of the program.

Generating the TIPC Session from the Log file

The values of the parameters used in the calculations and the messages that are displayed on the screen are also written to a log file TIPC.LOG. By printing this file, you can generate a written record of your session with TIPC. Each time your run TIPC, however, this file is overwritten; there is no attempt to archive the results of previous sessions.

3. Transionospheric Propagation

Display of parameter values

After you press the return key, information like the following will appear. The parameters displayed depend somewhat on the values of other parameters, such as the type of input pulse or the type of noise.

```
Enter ? for help, < to backup, ! to abort

Current Propagation Parameters:

     Type of input pulse     = Delta function
     Time resolution         =   1.0000E-03 (microseconds)
     Ionospheric TEC         =   1.0000E+00 (10**13/cm**2)
     Use ionospheric response calculated by TP
     Frequency range (LO,HI) =   3.6531E+01  5.0000E+02 (MHz)
     Type of noise           = none
     Hemisphere              = N
     Polarization mix        =   0.0000E+00 %
     Propagation angle       =   0.0000e+00 degrees
     Local oscillator freq.  =   0.0000E+00 (MHz)

     Choice of plots:
        Input pulse?              Y
        Input pulse spectrum?     Y
        Transionospheric signal?  Y
     Do not save input signal
     Do not save propagated signal

Are these values acceptable? <Y>
```

If you answer yes to this question, processing will continue. If you answer no, you will be given the chance to change any or all of the parameter values. In the latter case, the information will be displayed again with the new set of values.

Parameter descriptions

The following parameters are needed for transionospheric propagation:

type of input pulse

You have 5 types of input pulses to choose from. Here is an example of the menu that is presented to you.

```
Choose an input pulse type

::::::::::::::::::::::::::::::::::::::
::                                  ::
:: [1] Delta function               ::
:: [2] EMP: Double exponential      ::
:: [3] CPB:  Super Gaussian         ::
:: [4] Hyperbolic Tangent           ::
:: [5] Tabular pulse from a file    ::
:: [6] <backup to previous question> ::
:: [7] <abort this task>            ::
::                                  ::
```

```
:::::::::::::::::::::::::::::::::::::::::
Your choice? <1>
```

For each type of input pulse (1-5), you need to supply additional information.
[1] For delta function input, you must specify the number of delta functions along with the time delay and amplitude of each. The maximum number of delta functions allowed is controlled by the global parameter MAXDEL.
[2] For the double exponential input, you must specify alpha and beta parameters along with an amplitude and time delay.
[3] For the super Gaussian input, you must specify the amplitude, width, time delay, and exponent for the main pulse and for each reflection. The maximum number of reflections is controlled by the global parameter MAXRFL.
[4] For the hyperbolic tangent input, you need to specify the beam current, the 10% rise time, the 90% fall time, the pulse width, the beam velocity, the polar angle to the observer, and the distance to the observer. In addition you must specify the time delay and amplitude for each reflection. The maximum number of reflections is controlled by the global parameter MAXRFL.
[5] For tabular input, you need to specify the name of the file containing the data. For more details on this, see section 10.

Figure 1 displays the amplitude vs. time profiles for the default analytic input pulse forms. Figure 2 displays their Fourier transforms and Figure 3 displays their amplitude vs. time profile after being propagated through the default ionosphere.

time resolution

This is the time in microseconds between each sample value of the pulse.

transfer function

You may use a tabulated ionospheric transfer function. If you do not use a tabulated function, TIPC will create one from the input parameters.

ionospheric TEC

This is the value of the total electron content (TEC) of the ionosphere as a multiple of 10**13 electrons per square centimeter or 10**17 electrons per square meter.

Figure 4 shows how the TEC value strongly influences the transionopsheric signal for the case of the analytic super gaussian. The TEC was varied from 0.1 - 10. x 10^5 e/cm^2. We also note that the lower end of the frequency range changes from 11.6 to 112.8 MHz as the TEC value increases.

frequency range

Only frequencies within this range will be used in generating a time plot of the transionospheric signal.

effective plasma frequency

This is the value for the effective peak plasma frequency (fof2) corresponding to the TEC value specified.

type of noise

You have 4 types of noise for the input pulse to choose from. Here is an example of the menu that is presented to you.

```
Type of noise

: : : : : : : : : : : : : : : : : : : :
::                                   ::
:: [1] none                          ::
:: [2] white noise                   ::
:: [3] tabular noise                 ::
:: [4] scintillations                ::
:: [5] <backup to previous question> ::
:: [6] <abort this task>             ::
::                                   ::
: : : : : : : : : : : : : : : : : : : :
Your choice? <1>
```

[2] For white noise, you need to specify a signal-to-noise-ratio.
[3] For tabular noise, you need to specify the name of the file containing the data. For more information on this, see section 10.
[4] For scintillations, you need to specify a coherence frequency bandwidth.

An array of noise increments is constructed in the frequency domain. These values are added to the Fourier transform of the input signal before being propagated through the ionosphere. For choices [1] and [4], this array is set to zero. For choice [2], Gaussian random variables are generated with a mean of zero and a variance which is independent of the frequency and depends only the strength of the signal and the signal-to-noise-ratio. For choice [3], the tabulated values are used to generate Gaussian random variables with a mean of zero and variances that depend on the frequency. For choice [4], values of a special random function (see Roussel-Dupré and Kelley, 1990) are generated in the time domain and then its Fourier transform is used as an additional transfer function to be applied to the process of propagating the signal through the ionosphere.

Figure 5 displays four instances of white noise, with varying signal to noise ratios (SNR), added to the default electromagnetic pulse (EMP) signal. The input SNR value is the power value. The signal amplitude is being displayed and its SNR value is proportional to the square root of the power value. The white noise is added to the signal via a random function and therefore no two noise realizations will be identical.

hemisphere

Specify the hemisphere ('N' for northern or 'S' for southern) for which the calculations apply. This parameter affects the sign of the magnetic correction term in the ionospheric transfer function and effectively switches modes from extraordinary to ordinary and vice versa.

polarization mix

The percentage of extraordinary polarization to be used in simulating the deterministic effects of the ionosphere on the input signal. In the northern hemisphere, left-hand (LH) polarization is ordinary while right-hand (RH) polarization is extraordinary. Just the opposite is true in the southern hemisphere.

propagation angle

The angle in degrees between the propagation path of the signal and the earth's magnetic field.

local oscillator frequency

The frequency of a hypothetical oscillator used to shift the transionospheric signal. A transionospheric signal in tabular form may have been preprocessed in this manner.

plots

The program can generate a plot of the input pulse in the time domain, the input pulse in the frequency domain, the transionospheric signal in the time domain, or the tabular noise (if you specified that kind of noise). If you want any of these plots, you must answer Y to the appropriate question. If you ask to plot the input pulse on the time domain and if you have asked for noise to be added to the signal, then you will receive two plots - with and without the noise.

saving input signal

You can have TIPC save the time-domain representation of an input signal in a disk file. If you do, you must specify the name of the file.

saving propagated signal

You can have TIPC save the time-domain representation of the propagated signal in a disk file. If you do, you must specify the name of the file.

4. Signal Detection

Display of parameter values

After you press the return key, information like the following will appear.

```
Enter ? for help, < to backup, ! to abort

Current Signal Detection Parameters:

     Use propagated signal calculated by TP
     RCVR 1   Gaussian filter
           F0,DEL     =   1.5000E+02  2.0000E+00 (MHz)
           Polarization mix       =   0.0000E+00 %

     Choice of plots:
        Receiver amplitude spectrum?     Y
        Receiver phase spectrum?         Y
        Receiver time response?          Y
        Rcvd signal amplitude spectrum?  Y
        Rcvd signal phase spectrum?      Y
        Rcvd signal in time domain?      Y
        Power spectra?                   Y

Are these values acceptable? <Y>
```

If you answer yes to this question, processing will continue. If you answer no, you will be given the chance to change any or all of the parameter values. If you do so, then the information will be displayed again with the new set of values.

Parameter descriptions

The following parameters are needed for signal detection:

source for propagated signal

You can have TIPC use the signal values stored in a disk file as the propagated signal for further calculations.

number of receivers

The total number of receivers you want to define. The maximum is controlled by the parameter MAXRCVR defined in *tipc.inc*.

type of filter

You may choose from six types of filters: 1) a Gaussian filter for which you must specify the central frequency and a bandwidth, 2) a filter in tabular form (see section 10 for more details) for which you must specify a file name, or 3) a wideband filter for which you must specify a low and a high frequency cutoff and an order, or 4) a Butterworth filter for which you must specify a low and high frequency cutoff and an order, or 5) a Chebyshev type 1 filter for which you must specify a low and high frequency cutoff and an order, or 6) a Chebyshev type 2 filter for which you must specify a low and high frequency cutoff, an order, and the ratio between the rejection frequency and the cutoff frequency. A discussion of filter types can be found in Oppenheim and Schafer (1989).

saving receiver response

You can have TIPC store the frequency response of any receiver in a disk file. If you do, you must specify the name of the file.

saving received signals

You can have TIPC store the time-domain representation of the received signal in a disk file. If you do, you must specify the name of the file.

polarization

You must specify the percentage of extraordinary polarization that is detected by each receiver.

plots

The program can generate a plot of 1) the amplitude or 2) the phase of the receivers in the frequency domain, 3) the receiver response in the time domain, 4) the amplitude or 5) the phase of the received signals in the frequency domain, 6) the received signals in the time domain and 7) the power spectrum of the input pulse overlaid with an indication of what the receivers sample. If you want any of these plots, you must answer Y to the appropriate question.

Figure 6 displays the amplitude response of four types of filters. In each case the filter is centered at 150 MHz with a 2 MHz bandwidth. The oscillations in the transmission value in the passband of the Chebyshev type 1 filter are apparent.

5. Signal Processing

This task attempts to perform one of a number of signal processing algorithms to the received signals. When you choose this task, you must first go through the process of defining all the necessary parameters to the transionospheric propagation and signal detection tasks.

Display of parameter values

After you press the return key, information like the following will appear.

```
Enter ? for help, < to backup, ! to abort

Current Signal Processing Parameters:

     Use received signals computed by SD
     Signal processing method       = Square Law Detection
     Low-pass filter cutoff freq    =   1.0000E+01 (MHz)

     Choice of plots:
        Rcvd signal envelope?          Y

Are the values acceptable? <Y>
```

If you answer yes to this question, processing will proceed. If you answer no, you will be given the chance to change any or all of the parameter values. If you do so, then the information will be displayed again with the new set of values.

Parameter descriptions

The following parameters are needed for signal processing:

source for received signals

You can have TIPC use the signal values stored in disk files as received signals for further calculations.

signal processing method

Several methods of processing the signal are implemented in TIPC. Here is an example of the menu that will be presented to you.

```
Choose a signal processing method

::::::::::::::::::::::::::::::::::::::
::                                  ::
:: [1] Quadrature Filtering         ::
:: [2] Square Law Detection         ::
:: [3] Frequency Modulation TOA     ::
:: [4] Leading edge TOA             ::
:: [5] Cross correlation            ::
:: [6] Spectrogram                  ::
:: [7] <backup to previous question> ::
:: [8] <abort this task>            ::
::                                  ::
::::::::::::::::::::::::::::::::::::::
Your choice? <2>
```

[1] The quadrature filtering method pretends to sum the received signal with that generated by a local oscillator whose frequency you must specify. The in-phase and out-of-phase components of this sum are computed.
[2] The square law detection method passes the signal power through a low pass filter in order to compute the envelope of the received signal.
[3] The frequency-modulation time of arrival method first subjects the received signal to the square law operation. From the envelope, the peak region is determined. Then the received signal itself is examined in this region. The square of the difference between successive zeros is computed as a function of time. Then a straight line is fit to these data points. Thi^s line is extrapolated to the time axis. The point where it crosses the axis is defined to be the time of .rrival for the given filter.
[4] The leading edge time of arrival method first subjects the received signal to the square law operation to determine its envelope. The leading edge of the peak is examined to find where the peak drops to 2/3 and 1/3 of the peak value. A straight line through these two points is extrapolated to the time axis. The point where is crosses the axis is defined to be the time of arrival for the given filter.
[5] The cross-correlation method examines, as a function of s, the integral of $T_i(t)T_j(t\text{-}s)$ over t where T_i and T_j are the received signals for a pair of filters. The absolute value of s for which this function if maximum is defined to be the delta time of arrival for those two filters.
[6] The spectrogram uses a sliding time window to select subsets of the data as a nominal function of the time at the window's center. At present two windowing functions are available as options:

a "Parzen window"

$$w_j = \delta_{i-N_c-N/2,j}\begin{cases} 0, & j<0 \; or \; j>N, \\ 1-\left|\dfrac{j-\frac{1}{2}(N-1)}{\frac{1}{2}(N+1)}\right|, & 0\le j\le N, \end{cases}$$

and a "Welch window"

$$w_j = \delta_{i-N_c-N/2,j}\begin{cases} 0, & j<0 \; or \; j>N, \\ 1-\left(\dfrac{j-\frac{1}{2}(N-1)}{\frac{1}{2}(N+1)}\right)^2, & 0\le j\le N, \end{cases}$$

where i is the index to the original data set for which the spectrogram is being generated, j is the index to the resulting windowed data set, N_c is the index to the center of the time window in the original data set, and N is the number of points in the windowed data set. Note that the first point of the sets implicitly has an index of zero. The value of N is user selectable from a set of values that are integer powers of two. At present N_c steps in value from an initial value of $N/2$ in steps of N_s, to the highest value of m such that

$$mN_s + N \le N_d$$

where N_d is the number of data points in the original data, and $(m+1)$ is the number of windowed subsets of the original data. The windowed subsets are Fast Fourier transformed and the absolute value of the output complex values provides the spectral amplitude of the windowed data at the frequency values of

$$f_k = k/(N\Delta t), 0 \le k \le N/2+1$$

This $(m+1)\times(N/2+1)$ set of data can then be plotted using several different display options to show the spectral amplitude of the windowed data as a function of time of the window's center. This set of data can also be analyzed to determine, for a subset of frequency bins, the window center time at which the spectral amplitude of the bin is a maximum. The set of time versus frequency values is then plotted and used to estimate the ionospheric TEC, without, at present, corrections for detector dispersion, from the Delta Time of Arrival (DTOA) method discussed as part of the second level of signal processing.

estimation method

The program can estimate TEC and possibly the magnetic field correction factor based on the time of arrival data. For certain signal processing methods (frequency modulation, leading edge, and cross-correlation), you must specify the method of estimation to be used. This could be interpolation in a table of DTOA values, or a calculation based on the central frequencies of the receivers, or no estimation at all.

plots

The program can plot the quadrature results. It can plot the envelope of the filtered transionospheric signal. It can plot the squared period function calculated by the frequency modulation method. It can plot the envelopes used by the leading edge method. It can plot the cross correlation function mentioned above and it can plot a spectrogram. If you want any of these plots, you must answer Y to the appropriate question.

Figure 7 is an example of a spectrogram of a realistic input signal, entitled dish, to which narrowband FM signals have been added. The dish is from a Los Alamos transmitter and is an impulsive source. It is the boomerang shaped feature in the spectrogram. The horizontal bands centered at about 1 10 MHz are the FM signals. Significant aliasing is also apparent on the left side of the figure. Figure 8 is the tabular pulse amplitude vs. time before any FM is added.

6. Delta Time of Arrival Study

This task allows you to specify a number of values for TEC and then calculates the difference between the times of arrival for each pair of receivers for each value of TEC. When you choose this task, you must first go through the process of defining all the necessary parameters to the transionospheric propagation and signal detection tasks. You must also have defined at least two receivers when setting the signal detection parameters. You may choose one of these signal processing methods: FM, leading edge, or cross-correlation.

Display of parameter values

After you press the return key, information like the following will appear.

```
Enter ? for help, < to backup, ! to abort

Current DTOA study parameters:

     TEC range:               5.0000E-01  2.0000E+00
     Number of points           2

     Choice of plots:
        DTOA  vs. TEC?    Y
     Do not save DTOA tables

Are these values acceptable? <Y>
```

If you answer yes to this question, processing will continue. If you answer no, you will be given the chance to change any or all of the parameter values. If you do so, then the information will be displayed again with the new set of values.

Parameter descriptions

The following parameters are needed for the Delta Time of Arrival Study:

TEC range

You must specify the lower and upper values that will delineate the range of TEC values to be used. You may have the intermediate values interpolated automatically for you (on a log scale), or you may enter them yourself.

number of points

The number of TEC values to be used in the study. The maximum number is controlled by the global parameter MAXTEC.

plots

The program can plot the delta times of arrival as a function of TEC. If you want this plot, you must answer Y to this question.

Figures 9 and 10 display some of the results from a dtoa study using the provided parameter file *dtoa.prm*.

saving DTOA tables

You can have TIPC save in a disk file the tables that specify the differences in the arrival times for pairs of receivers as a function of TEC. If you do, you must specify the name of the file.

7. Delta Time of Arrival Uncertainty Study

Warning:
This study can be very time consuming and generate a large number of plots if the number of values and the number of realizations are large.

This task allows you to specify a number of values for the signal-to-noise ratio or the coherence frequency bandwidth and then estimates the mean and standard deviation of the difference in arrival times between each pair of receivers at each of those values. If you choose this task you must first specify the parameters needed for transionospheric propagation, signal detection, and signal processing. Again, you must have defined at least two receivers when setting the signal detection parameters.

Display of parameter values

After you press the return key, information like the following will appear.

```
Enter ? for help, < to backup, ! to abort

Current DTOA Uncertainty Parameters:

   range of coherence frequencies:  1.0000E+00  2.0000E+01 (MHz)
   Number of points             2
   Number of realizations        25
   Choice of plots:
     DTOA statistics?   Y

Are these values acceptable? <Y>
```

Parameter descriptions

The following parameters are needed for the Delta Time of Arrival Uncertainty Study:

range of values

You must specify the lower and upper values for the range of signal-to-noise ratios or coherence frequency bandwidths. You may have the intermediate values interpolated automatically for you (on a linear scale), or you may enter them yourself.

number of values

The number of values to be used in the study. The maximum number is controlled either by the global parameter MAXSNR or the global parameter MAXCOF.

number of realizations

The number of randomized cases to be generated for each specified value in the range of values. The code only requires that this number be a positive integer.

plots

The code can generate a plot of the sample mean and standard deviation of the delta times of arrivals vs. the specified values. This will be done for each pair of receivers.

Figure 11 displays the mean DTOA and its sigma for a study using the provided parameter file *delun.prm.* In this care 3 realizations of white noise were added to the default EMP pulse at two SNR values - 100 and 800.

8. Signal Reconstruction

If you choose this task you must first specify the parameters needed for signal detection and signal processing.

Display of parameter values

After you press the return key, information like the following will appear.

```
Enter ? for help, < to backup, ! to abort

Current Propagation Parameters:

   Estimate TEC by interpolation in DTOA table
   DTOA vs TEC file  = TIPCDTOA.DAT
   Plot reconstructed signal?     Y

Are these values acceptable? <Y>
```

Parameter descriptions

The following parameters are needed for signal reconstruction:

method of estimating TEC

You must choose between estimating TEC by interpolation in a table of DTOA values vs. TEC that had been previously constructed and saved or estimating TEC by calculation using values for the central frequency of each receiver.

DTOA table

If you choose the interpolation method of estimating TEC, you must specify the name of the file that contains the DTOA vs. TEC tables.

plots

The code can generate a plot in the time domain of the reconstructed input signal.

9. Saving and Restoring Parameter Values

Purpose

It does take some time for the user to define all the parameter values necessary for a certain task. To facilitate the reuse of a given set of parameter values, TIPC is capable of storing them in a file and at a later date reading them back in.

Format

The first two lines in the file are reserved for header and alignment information. The remaining lines contain parameter value settings, one value for each line. The format is name of parameter followed by equal sign followed by its value. The name of the parameter is the name of the variable that is used in the program. Here is a listing of such a file:

```
NAME             VALUE
---------------  ----------
AMP0           =  1.000000E+00
BV             =  9.990000E-01
C2             =  8.000000E+01
CI             =  1.000000E+01
COFH           =  2.000000E+01
COFL           =  1.000000E+00
D2             =  1.000000E+02
DLT0           -  4.000000E-03
DT             =  1.000000E-03
EMAX           =  5.000000E+02
F1             =  3.653101E+01
F2             =  5.000000E+02
FCOH           =  1.500000E+02
FLCO           =  1.000000E+01
HEMISPHERE     = N
INOISE         =         1
IPFILE         = TIPCIP.DAT
IPS            =         4
IPULSE         =         1
IRFL1          =         2
IRFL2          =         1
ITEC           =         1
LOF            =  1.500000E+02
NCOF           =         2
NDEL           =         1
NRCVR          =         3
NRZNS          =        25
NSFILE         = NOISE.DAT
NSNR           =         2
NTEC           =         2
PLDTOA         = Y
PLDTU          = Y
PLFIP          = N
PLFIPP         = N
PLFRTIS        = N
PLFSR          = N
PLFTN          = Y
PLTCC          = Y
```

```
PLTIP           = N
PLTLE           = N
PLTLPFS         = Y
PLTPSQ          = Y
PLTQUAD         = Y
PLTREC          = Y
PLTRTIS         = N
PLTSR           = N
PLTTIS          = N
PMIX            =  0.000000E+00
RR              =  2.500000E+05
SAVEDT          = N
SAVEIP          = N
SAVETS          = N
SNR             =  1.000000E+02
SNRH            =  1.000000E+02
SNRL            =  1.000000E+00
TEC             =  1.000000E+00
TECH            =  2.000000E+00
TECL            =  5.000000E-01
TF              =  2.000000E-02
THETA           =  9.000000E+01
TR              =  1.000000E-02
TSFILE          = TIPCTS.DAT
TT0             =  3.000000E-02
TW              =  5.000000E-02
USERS           = Y
USETS           = N
XN0             =          4
XZR             =  5.000000E-01
TDEL(1)         =  4.000000E+00
YNOR(1)         =  9.999999E+02
AMP1(1)         =  3.000000E-01
DLT1(1)         =  4.000000E-03
TT1(1)          =  4.000000E-02
XN1(1)          =          4
AMP1(2)         = -2.000000E-01
DLT1(2)         =  4.000000E-03
TT1(2)          =  5.000000E-02
XN1(2)          =          4
ARFL(1)         = -1.000000E+00
TRFL(1)         =  1.000000E-02
FF0(1)          =  1.500000E+02
FDEL(1)         =  2.000000E+00
IRCVR(1)        =          2
SRFILE(1)       = TIPCF01.DAT
FLOW(1)         =  0.000000E+00
FHIGH(1)        =  0.000000E+00
RSFILE(1)       = TIPCS01.DAT
SAVERS(1)       = N
SAVESR(1)       = N
FF0(2)          =  1.500000E+02
FDEL(2)         =  2.000000E+00
IRCVR(2)        =          2
SRFILE(2)       = TIPCF02.DAT
FLOW(2)         =  0.000000E+00
FHIGH(2)        =  0.000000E+00
```

```
RSFILE(2)          = TIPCS02.DAT
SAVERS(2)          = N
SAVESR(2)          = N
FF0(3)             =  1.500000E+02
FDEL(3)            =  2.000000E+00
IRCVR(3)           =           2
SRFILE(3)          = TIPCF03.DAT
FLOW(3)            =  0.000000E+00
FHIGH(3)           =  0.000000E+00
RSFILE(3)          = TIPCS03.DAT
SAVERS(3)          = N
SAVESR(3)          = N
COFVEC(1)          =  1.000000E+00
COFVEC(2)          =  2.000000E+01
SNRVEC(1)          =  1.000000E+00
SNRVEC(2)          =  1.000000E+02
TECVEC(1)          =  5.000000E-01
TECVEC(2)          =  2.000000E+00
```

Specifying a file

If you choose to save parameter values, you will be asked for the name of a file (the default is TIPC.PRM). If this file already exists, you will be asked if you want to overwrite it. Then the values of the major parameters will be written to this file in the format that is illustrated above. Since this is a text file, you could use an editor to change any of the values in it.

If you choose to restore parameter values, you will be asked for the name of a file (the default is TIPC.PRM). Then the program will attempt to translate all the information in that file into parameter values. These are the values that would then be displayed after you choose one of the first 6 tasks.

10. Using Tabular Data

The TIPC program is able to accept a variety of tabular data as input. First of all the file must be a text file capable of being edited with a standard editor. Secondly, the first line or two in the file must contain an *information message*. that consists of one or more pieces of entries that has the format: keyword followed by equal sign followed by a "value" followed by a semicolon. Such a message supplies all the needed information for the code to extract the data that follows.

Here is an example of such a data file:

```
NPTS = 3; TYPE = (T,A); FORMAT = (1P2E15.6); DELAY =   4.000000E+00;
  3.999000E+00  0.000000E+00
  4.000000E+00  9.999999E+02
  4.001000E+00  0.000000E+00
```

The following sections will explain the keywords that are recognized for these types of tabular input: a) the input signal, b) the tabular noise, c) the propagated signal, d) the frequency response for a receiver, e) a signal received by a receiver, and f) the DTOA vs. TEC tables.

The provided file *dishnu* is an example of a tabulated input signal.

Input signal

The keywords that are recognized are
NPTS - the number of data points defining the function,
TYPE - specifies how the values are presented,
FORMAT - format that can be used in reading the values,
DELAY - the delay in microseconds associated with the input signal,
ASCALE - the value to scale the amplitudes by to get volts/meter,
TSCALE - the value to scale the times by to get microseconds.
For now, the only allowed value for TYPE is (T,A) meaning that the signal values come in pairs with time being the first number and amplitude being the second number.

Here is an example:

```
NPTS = 5001;  TYPE = (T,A);  FORMAT = (1P2E16.6);
```

This is the minimum amount of information that you must specify. Default values are provided for ASCALE, TSCALE, & DELAY (1.0, 1.0, & 0.0 respectively) if you do not specify them. Note that all the parentheses and all the semicolons in the example above are required.

Noise

The keywords that are recognized are
NPTS - the number of data points,
FORMAT - the format used to read the data values.
The data must be in the form of triples. The first number must the frequency value. The second number must be the mean value of the random noise to be applied. The third value must the standard deviation of the random noise to be applied. A Gaussian random value with this mean and this standard deviation will be added to the spectrum of the input signal.

Here is an example:

```
NPTS = 500; FORMAT = (1P3E16.6);
```

Note that all the parentheses and all the semicolons in this example are required.

Propagated signal

The keywords that are recognized are
NPTS - the number of data points defining the function,
TYPE - specifies how the values are presented,
FORMAT - format that can be used in reading the values,
DELAY - the delay in microseconds associated with the signal,
ASCALE - the value to scale the amplitudes by to get volts/meter,
TSCALE - the value to scale the times by to get microseconds,
LOF - the local oscillator frequency used to "preprocess" the signal,
PMX - the percent of extraordinary polarization in the signal,
MFCF - the magnetic field correction factor.
For now, the only allowed value for TYPE is (T,A) meaning that the signal values come in pairs with time being the first number and amplitude being the second number.

Here is an example:

NPTS = 5001; TYPE = (T,A); FORMAT = (1P2E16.6);

This is the minimum amount of information that you must specify. Default values are provided for ASCALE, TSCALE, DELAY, LOF, PMX & MFCF (1.0, 1.0, 0.0, 0.0, 0.0, & 0.8 respectively) if you do not specify them. Note that all the parentheses and all the semicolons in the example above are required.

Receivers

The keywords that are recognized are
NPTS - the number of data points,
TYPE - specifies how the data is presented,
FORMAT - the format to be used in reading the data,
DELAY - the delay in microseconds associated with the receiver,
FC - the central frequency associated with the receiver,
FW - the frequency width associated with the receiver.

The data must be in the form of triples. The first value is the frequency. The other two specify a complex number indicating the action of the receiver at that frequency. If the complex number is given in rectangular form, (F,X,Y) must be the value for TYPE. If the complex number is given in polar form, (F,A,P) must be the value for TYPE.

Here is an example:

NPTS = 500; TYPE = (F,A,P); FORMAT = (1P3E16.6);

This is the minimum amount of information you must supply. Default values for DELAY, FC, & FW (0., 0., & 0. respectively) are provided if you do not specify them. Note that all the parentheses and all the semicolons in the example above are required.

Received signals

The keywords that are recognized are
NPTS - the number of data points defining the function,
TYPE - specifies how the values are presented,
FORMAT - format that can be used in reading the values,
ASCALE - the value to scale the amplitudes by to get volts/meter,

TSCALE - the value to scale the times by to get microseconds,
DELAY - the delay in microseconds associated with the each signal.
For now, the only allowed value for TYPE is (T,A) meaning that the signal values come in pairs with time being the first number and amplitude being the second number.

Here is an example:

NPTS = 5001; TYPE = (T,A); FORMAT = (1P2E16.6);

This is the minimum amount of information that you must specify. Default values are provided for ASCALE, TSCALE, & DELAY (1.0, 1.0, & 0.0 respectively) if you do not specify them. Note that all the parentheses and all the semicolons in the example above are required.

DTOA tables

The keywords that are recognized are
NTEC - the number of TEC values,
NRCVR - the number of receivers,
MFCF - the magnetic field correction factor, &
FORMAT - format that can be used in reading the values.

Here is an example:

NTEC = 25; NRCVR = 3; FORMAT = (2I4,1P4E15.6);

Note that all the parentheses and all the semicolons in the example above are required.

11. Subroutine Descriptions

The Fourier transform routines CFFTB, CFFTF, and CFFTI were taken from the CLAMS (Common Los Alamos Mathematical Software) library. They call several subroutines that are not described here. Also, only the higher level calls that are actually used from the OINK library are described.

AMAXMIN (ARRAY, N, AMAX, IMAX, AMIN, IMIN)

This subroutine in *disp.f* or *ncar.f* searches the first N elements of the array ARRAY for the maximum and minimum value. It returns the maximum value in AMAX and its corresponding index in IMAX. It also returns the minimum value in AMIN and its corresponding index in IMIN.

ANNOUNCE (MSG)

Display an announcement message contained in MSG in the log file and on the user's terminal.

BANDPASS (FLO, FHI, COEF, FLTR)

This subroutine constructs a wide band filter by smoothly patching together exponential functions in such a way that there is 50% transmission at the specified low and high frequencies of FLO and FHI. COEFF governs the steepness of the exponentials. The resulting complex function is returned in FLTR.

BUTTER (FLO, FHI, ORDER, FLTR)

This subroutine constructs a Butterworth filter of order ORDER such that frequencies between FLO and FHI are passed and there is 50% power gain at these two frequencies. The resulting complex function is returned in FLTR.

CFFTB (N, C, WSAVE)

This subroutine in fft.f computes the backward complex (discrete) Fourier transform of the sequence of N points in the array C. WSAVE is a work array that must be first initialized with a call to CFFTI.

CFFTF (N, C, WSAVE)

This subroutine in fft.f computes the forward complex (discrete) Fourier transform of the sequence of N points in the array C. WSAVE is a work array that must be first initialized with a call to CFFTI.

CFFTI (N, WSAVE)

This subroutine in fft.f initializes values in the work array WSAVE for use with CFFTB and CFFTF.

CHEBY1 (FLO, FHI, ORDER, FLTR)

This subroutine constructs a Chebyshev type 1 filter of order ORDER such that frequencies between FLO and FHI are passed and there is 50% power gain at these two frequencies. The resulting complex function is returned in FLTR.

CHEBY2 (FLO, FHI, ORDER, WRWC, FLTR)

This subroutine constructs a Chebyshev type 2 filter of order ORDER (which must be odd) such that frequencies between FLO and FHI are passed, there is 50% power gain at these two frequencies, and the ratio between the rejection frequency and the cutoff frequency is WRWC. The resulting complex function is returned in FLTR.

CIRF (CDT, DD, NTIM, CR)

This subroutine computes NTIM values of a random function in the time domain which is meant to simulate the stochastic features of the ionosphere. The Fourier transform of this function is returned in CR. CDT is the time increment to be used while DD is the delay interval length to be used. The coherence frequency bandwidth, FCOH, is also used in constructing the function.

COLORBAR (NCOL, RLEV, ITYPE, TITL)

This subroutine draws a colored barstrip at the top of a XYZ plot. NCOL is the number of colors with numeric equivalents stored in the array RLEV(NCOL).
ITYPE indicates whether a linear (ITYPE=1) axis or logarithmic (ITYPE=2) axis should be drawn. TITL is the label for the colorbar.

CONTOUR (W, XV, YV, NXP, NYP, WMIN, WMAX, CON, NCON, XMIN, XMAX, YMIN, YMAX, XLABEL, YLABEL, IOPT, IOPTZ)

This subroutine draws a contour plot of the array W(XV,YV). WMIN is the minimum W value and WMAX is the maximum value. XV is dimensioned NXP and YV is dimensioned NYP. The number of contours to be drawn in NCON and their values are stored in the CON(NCON) array. IOPT indicates whether a linear-linear, linear-log, log-linear, or log-log axis setup should be used. IOPTZ is currently unused.

CROSCOR (INDEX)

This subroutine estimates the difference in times of arrivals for pairs of receivers by determining where the integral of $T_i(t)T_j(t\text{-}s)$ over t is maximized (T_i and T_j are the received signals for receivers i and j). This estimate is placed in the array DTOA using INDEX as a guide. This subroutine can also generated a plot of the above integral depending on the flag PLTCC.

DATETIME (STRING)

This subroutine returns the current date and time formatted in STRING.

ESTTEC (INDEX, TECVAL, MFCFVAL)

Estimate the value of TEC and the magnetic field correction factor, MFCF, by either interpolation or calculation (using central frequencies) depending on the value of ITEC. Return the values in TECVAL and MFCFVAL. The subroutine LLSQ2 is called to perform the least squares analysis if calculation is chosen.

ETTYDEF

This is a dummy routine for the proper functioning of the OINK library.

ETTYNOP

This is a dummy routine for the proper functioning of the OINK library.

FFT (NT, NF, DT, TVAL, CFT, WORK)

This subroutine computes the forward Fourier transform of a function. NT values of the function in the time domain are taken from TVAL (where DT is the time increment between successive values) and converted to NF values in the frequency domain. The resulting complex spectrum is output in CFT. WORK is used as a scratch array.

FILTERS

This subroutine establishes the necessary information to define the receivers. The time domain responses are stored in TVAL. The complex Fourier transforms are stored in FIL. The magnitudes of the spectral values are stored in FVAL. Time delays to be associated with each receiver are stored in RCVRDLY.

FMTOA (VAL1, VAL2, NPTS, TIME, TOA, LEFT, RIGHT)

This subroutine estimates the time of arrival for a receiver by the Frequency Modulation method. It uses the envelope of the received signal in VAL1 to define a peak region. The boundaries of this region are returned in LEFT and RIGHT. The actual received signal in VAL2 is examined over this region. A function which is the square of the difference between successive zeros of the signal versus the time values in TIME is constructed in VAL2 (overwriting the previous values). NPTS is the number of points in arrays VAL 1 and VAL 2. A straight line is then fit to this function. The point where this line crosses the time axis is returned in TOA as the time of arrival for that receiver.

FRANGE (IESC)

This subroutine determines the frequency range to be used in generating time domain plots so that the plotted signal will fit within the time window allotted. IESC is the escape code entered by the user.

FREQPLOT (YVAR, IST, ND, NCURVES, NPLOTS, IOPT, SNGL, USRLBL)

This subroutine generates plots in the frequency domain. YVAR is the array of values to be plotted. IST is the starting index for each column of the array while ND is the number of values to plot for each of the NPLOTS. NCURVES is the number of curves to overplot on each graph. IOPT indicates the type of plot desired. SNGL is a flag indicating whether single plots per page are requested. USRLBL contains the user specified label, if any, for the plot.

GCLR (X)

This subroutine sets the color to 1 of 129 colors from pure blue to pure red as X varies from zero to one.

GETDTOA (MCF)

Retrieve the DTOA vs. TEC tables stored in the file whose name is found in DTFILE and place them in the array DTOA. MCF is the magnetic field correction factor.

GETIRF (NAME, CARAY1, CARAY2, ERROR)

This subroutine opens a file containing the frequency-domain representation of the ionospheric response function, parses an information message, and fetches the data. NAME is the name of the file containing the data; CARAY1 is the ordinary ray response function; CARAY2 is the extra-ordinary ray response function, and ; ERROR is a logical flag indicating any error.

GETMSG (LUN)

Retrieve the information message from the beginning of a data file using the logical unit number specified in LUN. This message is contained in the first one or two records of that file.

GETNOISE

This subroutine retrieves noise data from a file whose name is given in NSFILE and then sorts the values.

GETNP (FR, DM, DS)

This subroutine calculates the noise parameters DM and DS corresponding to a given frequency FR by interpolating in the random noise table.

GETPARAM (LUN, NAME, VAL, EOF, ERROR)

This subroutine retrieves a record from a parameter file using the logical unit number LUN. The parameter name is returned in NAME and its value is returned in VAL. The logical flag EOF indicates whether the end of the file has been reached. The logical flag ERROR indicates whether an error occurred while reading the file or whether the record has the wrong syntax.

GETRCVR (NR)

This subroutine retrieves tabular data from a file whose name appears in SRFILE and which defines the frequency response of a receiver. NR is the index of the receiver.

GETREGS

This subroutine constructs nonoverlapping regions where the receivers have transmissions exceeding .001.

GETSGNL (NAME, ARRAY, DELAY, OSCFRQ, PMX, MCF, ERROR)

This subroutine retrieves a time-domain representation of a signal from a disk file whose filename is stored in NAME. It parses the information message returning any parameters associated with the signal. Then it calls READSGNL to retrieve the actual data which is stored in ARRAY. The signal delay is returned in DELAY. The associated local oscillator frequency is returned in OSCFRQ. The percentage of extraordinary polarization in the signal is returned in PMX. The associated magnetic field correction factor is returned in MCF. ERROR is a logical flag that is set to time when the file cannot be opened or read , or when information in the file cannot be decoded.

GETSPEC (NAME, ARRAY, DELAY, FC, FW, ERROR)

This subroutine retrieves a frequency-domain representation of a signal from a disk file whose filename is stored in NAME. It parses the information message returning any delay associated with the signal in DELAY, any central frequency associated with the signal in FC, and any frequency width associated with the signal in FW. Then it calls READSPEC to retrieve the actual data which is stored in ARRAY.

GETTASK (ITASK, FLAG1, FLAG2, FLGA3)

This subroutine requests from the user the task that is to be performed (returned in ITASK) and calls the appropriate menu routines to establish the necessary parameter values. FLAG1, FLAG2, and FLAG3 indicate whether parameters were modified by the menus corresponding to Transionospheric Propagation, Signal Detection, and Signal Processing respectively.

GRPHINIT

This routine initializes the graphics routines.

INITMCF

This subroutine initializes the magnetic field correction factor based on the hemisphere chosen and the angle between the pulse direction and the magnetic field.

INITMENU

This block data subprogram initializes the various menu options.

INITPLTS

This block data subprogram initializes the data concerned with the plot options.

INITTF

This subroutine initializes the array of times in TIME and frequencies in FREQ. It uses the time increment DT and the number of samples points NPTS to do this. It also defines TMAX, TMID, and FMAX.

INITVAL

This subroutine performs the necessary initialization when TIPC is started up.

INITVEC (ARRAY, LOW, HIGH, NUM, LOGSCALE)

This subroutine establishes an array of NUM values in ARRAY by interpolating between the low and high values (LOW and HIGH respectively). LOGSCALE indicates whether interpolation is to be done on a log scale or not.

INTERP (N, ARRAY1, ARRAY2, VAL1, VAL2, ERROR)

This subroutine performs a linear interpolation. It determines the position of VAL1 among the N elements of ARRAY1 and then interpolates in ARRAY2 returning the result in VAL2. ERROR is a logical variable which is set to true if VAL1 does not fall within the range of values in ARRAY1.

IOFIRS (CIN)

This function in the OINK library determines the position of the first significant character in the string CIN.

IOGETC (CPROMP, CESCAP, CINOUT, IESC)

This subroutine in the OINK library presents the prompt in CPROMP to the user and accepts the user's input of a string in CINOUT. CESCAP contains a string of 'escape' characters. The status of the user's input is returned in IESC.

IOGETI (NEXPEC, CPROMP, CESCAP, IINOUT, NFOUND, IESC)

This subroutine in the OINK library presents the prompt in CPROMP to the user and accepts the user's input of an integer value or array in IINOUT. NEXPEC is the number of values to be expected. CESCAP contains a string of 'escape' characters. The number of values entered is returned in NFOUND. The status of the user's input is returned in IESC.

IOGETR (NEXPEC, CPROMP, CESCAP, RINOUT, NFOUND, IESC)

This subroutine in the OINK library presents the prompt in CPROMP to the user and accepts the user's input of a real value or array in RINOUT. NEXPEC is the number of values to be expected. CESCAP contains a string of 'escape' characters. The number of values entered is returned in NFOUND. The status of the user's input is returned in IESC.

IOGFIL (CPROMP, CESCAP, CSTAT, CFNAME, IESC)

This subroutine in the OINK library presents the prompt in CPROMP to the user and accepts the user's input of a file name in CFNAME. The string in CSTAT is then used to determine the file's status. CESCAP contains a string of escape characters. The status of the user's input is returned in IESC.

IOINIT (CSYS, CASE, NWIDTH)

This subroutine in the OINK library initialize the internal variables that are system dependent or global in nature. This must be called before any other OINK subroutine in invoked. CSYS contains the operating system keyword (vms, ctss, or unix). CASE indicates the case in which character information is returned to the program from some of the OINK routines (u or l). NWIDTH defines the width of the terminal screen or window in characters.

IOLAST (CIN)

This function in the OINK library determines the position of the last significant character in the string CIN.

IONMEN (NITEM, CTITLE, NEXPEC, CPROMP, IOPT, NFOUND)

This subroutine in the OINK library handles menus with integer labelled options. NITEM is the number of items to appear on the menu. CTITLE contains the title to appear above the menu (may be a null string). NEXPEC indicated the number of options that the user is expected to enter. The array of prompts for the menu items is in CPROMP. The user's choices are returned in IOPT. The number of menu items chosen is returned in NFOUND.

IOOFIL (LUNIT, CPROMP, CESCAP, CSTAT, CFNAME, IESC)

This subroutine in the OINK library prompts the user with CPROMP and accepts the user's input of a file name in CFNAME. The string in CSTAT is used as the status in attempting to open the file using the logical unit number specified in LUNIT. CESCAP contains a string of 'escape' characters. The status of the user's reply is returned in IESC.

IOUSER

This is a dummy routine for the proper functioning of the OINK library.

IOYORN (CPROMP, CESCAP, CYORN, IESC)

This subroutine in the OINK library presents the prompt in CPROMP to the user in the form of a yes or no question and accepts the user's response in CYORN. CESCAP is a string of 'escape' characters. The status of the user's reply is returned in IESC.

IPCPB1

This subroutine generates a supergaussian input signal.

IPCPB2

This subroutine generates a hyperbolic tangent input signal.

IPDELTA

This subroutine generates a set of delta functions for an input signal.

IPEMP

This subroutine generates double exponential function for an input signal.

IPROP

This subroutine simulates the effect of the ionosphere on the input signal. It first adds the noise requested to the input signal. Then it applies to the Fourier transform of the input signal a transfer functions which includes phase shifts as a function of frequency in order to simulated the deterministic aspects of the ionosphere. If scintillations are requested, a stochastic transfer function is generated which is also applied to the signal. This subroutine can generate a plot of the transionospheric signal if indicated by PLTTIS.

ITRANS

This subrountine obtains the ionospheric transfer function.

LABEL1D (STRING, L1, IV)

This routine generates labels for PLOTXY and PLTXYZ. The label is entitled STRING and is of length L1. IV ranges from 1 to 5 and indicates where on the plot the label is positioned.

LETOA (ARRAY, NPTS, TIME, TOA, LEFT, RIGHT)

This subroutine estimates the time of arrival for a receiver by the leading edge method. The envelope of the NPTS values of the received signal is in the array ARRAY. Two points on the leading edge where the signal drops to 2/3 and 1/3 of the peak value are found. A straight line is drawn between these two points, and the point where the line intersects the time axis is declared to be the time of arrival for that receiver. The time values in TIME are used for this purpose. The time of arrival is returned in TOA. Also returned are boundaries of the signal peak region, LEFT and RIGHT, for plotting purposes.

LOGIT (MSG)

This subroutine writes the character string contained in MSG to the user's terminal and into the log file.

LLSQ1 (N, X, Y, CF, EST1, SIG1, SSE, IERR)

This subroutine calculates the least squares estimates of a in the following model: $y_i = a(x_i + CF)$. N is the number of data points in the X and Y arrays. The estimate of a and its standard error are returned in EST1 and SIG1. The chi-square value is returned in SSE. Any error conditions are returned as negative values in IERR.

LLSQ2 (N, X, Y, EST1, EST2, SIG1, SIG2, SSE, IERR)

This subroutine calculates the least squares estimates of a and b in the following model: $y_i = a(x_i + b)$. N is the number of data points in the X and Y arrays. The estimate of a and its standard error are returned in EST1 and SIG1. The estimate of b and its standard error are returned in EST2 and SIG2. The chi-square value is returned in SSE. Any error conditions are returned as negative values in IERR.

MENUDT (FLAG, IESC)

This subroutine requests from the user all the necessary information needed to perform a delta time of arrival study. FLAG indicates whether any of the parameter values were changed. IESC is the escape code corresponding to the user's last response.

MENUDU (FLAG, IESC)

This subroutine requests from the user all the necessary information needed to perform a delta time of arrival uncertainty study. FLAG indicates whether any of the parameter values were changed. IESC is the escape code corresponding to the user's last response.

MENUPS (FLAG, IESC)

This subroutine requests from the user all the necessary information needed to perform signal processing. FLAG indicates whether any of the parameter values were changed. IESC is the escape code corresponding to the user's last response.

MENUSD (FLAG, IESC)

This subroutine requests from the user all the necessary information needed to perform signal detection. FLAG indicates whether any of the parameter values were changed. IESC is the escape code corresponding to the user's last response.

MENUSR (FLAG, IESC)

This subroutine requests from the user all the necessary information needed to perform signal reconstruction. FLAG indicates whether any of the parameter values were changed. IESC is the escape code corresponding to the user's last response.

MENUTP (FLAG, IESC)

This subroutine requests from the user all the necessary information needed to perform transionospheric propagation. FLAG indicates whether any of the parameter values were changed. IESC is the escape code corresponding to the user's last response.

NEWSEED (SEED)

This subroutine uses the current system time to establish an initial value for SEED which is used in the generation of random numbers.

NOISE

This subroutine defines the noise that is to be added to the input signal. It can also generate plots of the tabular noise, the input signal plus noise, or the spectrum of the input signal plus noise.

NOISE1

This subroutine generates white noise for the input signal.

NOISE2

This subroutine generates frequency depended noise for the input signal based on the tabulated data in a file. The name of the file is stored in NSFILE.

NOPRMT

This is a dummy routine for the proper functioning of the OINK library.

PARAMS (MODE)

This subroutine either saves the parameter values in a file or restores those values from a file depending on the string in MODE. The user is prompted for the name of the parameter file.

PARSEMSG (MSG, KEYWORD, VALUE, IESC)

This subroutine extracts the keyword and value from the first segment of an information message in MSG which is included with tabular data. These are returned in KEYWORD and VALUE respectively. The messages minus the first segment is returned in MSG. An error condition is return in IESC.

PAUSE

This subroutine prints out an innocuous message and waits for any response from the user.

PLOTEXIT (TTY, STTY)

This subroutine in terminates any graphics output.

PLOTINIT (TTY, STTY)

This subroutine prepares for graphics output.

PLOTXY (X, Y, ISTART, NDIM, NPTS, NC, IPLT, NPLT, IOPT, XLABEL, YLABEL, TITLE0, TITLE1, TITLE2, MESSAGE, TTY)

This subroutine produces one or more X-Y plots. The columns of Y are plotted as a function of X starting at ISTART. NDIM is the column dimension of the two dimensional array Y. NPTS is the number of points to plot. NC is the number of curves to plot per graph. IPLT is the index of the plot (from 1 to 4) that indicates where on the page it is to be plotted in case there is more than one plot. NPLT is the total number of plots. IOPT indicates the type of scales to use (1 for linear X, linear Y; 2 for linear X, logarithmic Y; 3 for logarithmic X, linear Y; and 4 for logarithmic X, logarithmic Y). XLABEL contains the label for the X axis while YLABEL contains the label for the Y axis. TITLE0, TITLE1, and TITLE2 contain titles and subtitles to be displayed at the top of the page. MESSAGE contains secondary information also to be displayed immediately below the titles. TTY indicates the type of terminal being used.

PLTQUERY (INDEX, IESC)

This subroutine queries the user for plot titles and whether single plots per page are desired. INDEX points into the array of plot flags, prompts, and titles. The escape code for the user's last response is returned in IESC.

PLTXYZ (X, Y, Z, ZRANG0, ZSTEP0, IXSTRT, JYSTRT, NXDIM, NYDIM, NXPTS, NYPTS, IPLT, NPLT, IOPTXY, IOPTZ, XLABEL, YLABEL, QUANLAB, ZUNLAB, TITLE0, TITLE1, TITLE2, MESSAGE, TTY)

This subroutine generates one or more plots of Z versus X and Y. ZRANG0 is the range of values for ZVAL used in generating contours. ZSTEP0 is the step size for contours of ZVAL. The rows of Z are plotted as a function of X starting at IXSRT, and the columns of Z are plotted as a function of Y starting with JYSTRT. NXDIM and NYDIM are the row and column dimensions of the three dimensional matrix Z. NXPTS and NYPTS are the number of X and Y values associated with the set of Z data used in the plots. IPLT is the index of the plot (from 1 to 4) that indicates the plot location in case there is more than one plot. NPLT is the total number of plots. IOPTXY indicates the type of scales to use for the axes (1 for linear X, linear Y; 2 for linear X, logarithmic Y: 3 for logarithmic X, linear Y; and 4 for logarithmic X, logarithmic Y). IOPTZ indicates the type of scales to use for contours of Z (1 for linear contour plot of Z; 2 for decibel contour plot of Z;3 for linear false color plot of Z, and 4 for decibel false color plot of Z). XLABEL is the label for the X axis. YLABEL is the label for the Y axis. QUANLAB is the label identifying the variable plotted as Z. ZUNLAB is the label identifying the units of the Z variable. TITLE0, TITLE1, and TITLE2 are the user supplied title, main title, and sub-title respectively. MESSAGE is an additional header message. TTY identifies the terminal type.

POLY (N, COEF, X)

This function returns the value of a polynomial at X which is specified by N coefficients stored in COEF.

PULSE (IESC)

This subroutine requests from the user the information needed to determine the input signal. Returned in IESC is the escape code corresponding the user's last response.

QUAD

This subroutine performs the quadrature filtering algorithm. It sums the received signal itself and the received signal shifted out of phase by 90 degrees with a fictitious local oscillator. It can plot these quadrature results if indicated by PLTQUAD.

QUADFILL (W, XV, YV, NXP, NYP, WMIN, WMAX, CON, NCON, XMIN, XMAX, YMIN, YMAX, XLABEL, YLABEL, IOPT, IOPTZ)

This routine fills quadrilaterals in the XV-YV plane with a color corresponding to the average value of W within the quadrilateral. XMIN andXMAX are the limits of the XV axis labeled XLABEL. YMIN and YMAX are the limits of the YV axis labeled YLABEL. WMIN is the minimum W value and WMAX is the maximum value. XV is dimensioned NXP and YV is dimensioned NYP. The number of colors to be drawn is NCON and their numeric equivalents are stored in the CON(NCON) array. IOPT indicates whether a linear-linear, linear-log, log-linear, or log-log axis setup should be used. IOPTZ is currently unused.

RANDG (MEAN, SIGMA, SEED)

This subroutine generates Gaussian random values with mean and standard deviation specified by MEAN and SIGMA respectively. SEED is automatically updated.

RANDU (SEED)

This subroutine generates uniform random values between zero and one. SEED is automatically updated.

READTF (CARRAY, NF, TYPE, FSCALE, ERROR)

This subroutine reads the frequency-domain values and places them in the appropriate position of the response function array, CARRAY. NF is the number of frequency points, TYPE is a character array that specifies how the data is presented, FSCALE is the value by which to scale the frequencies in order to obtain MHZ, and ERROR is a logical flag that is set to true if TYPE is not equal to (F,A,P), then an error message is given.

READTF2 (CARAY1, CARAY2, NF, TYPE, FSCALE, ERROR)

This subroutine reads the frequency-domain values and places them in the appropriate position of the response function arrays, CARAY1 and CARAY2.

READSGNL (ARRAY, NT, TYPE, TSCALE, ASCALE, ERROR)

This subroutine reads NT values from the data file using the logical unit LUNXFR and stores the results in ARRAY. The character string TYPE indicates how the data is structured. TSCALE and ASCALE are used to scale the time and/or the amplitudes of the signal.

READSPEC (ARRAY, NF, TYPE, ERROR)

This subroutine reads NF values from the data file using the logical unit LUNXFR and stores the results in ARRAY. The character string TYPE indicates how the data is structured. Error is the logical flag that is set to true if TYPE does not correspond to the three allowed values, there is an error message.

RECSGNL (TECVAL)

This subroutine reconstructs the FFT of the input signal by using the nonoverlapping regions where the receivers are "active". In each region, the FFT of the received signal is divided by the transfer functions for the receiver. This is then divided by the transfer function for the ionosphere (TECVAL is used as the value for TEC in creating this). The result is an estimate of the FFT for the input signal. Then the inverse Fourier Transform is applied to determine an estimate of the input signal itself.

RESTORE (NAME, INDX, VAL)

This subroutine restores the value of a parameter specified in a line of the parameter file. The parameter's name if given in NAME. If it is an array, the index is given in INDX. The character string containing the value is given in VAL.

SAVEC (LUN, NAME, VARIABLE)

This subroutine encodes the value of a character string variable stored in VARIABLE along with its name in NAME and writes this out to the parameter file using the logical unit number specified in LUN.

SAVEDTOA

This subroutine saves the DTOA vs. TEC tables in a file.

SAVEI (LUN, NAME, VARIABLE)

This subroutine encodes the value of an integer variable stored in VARIABLE along with its name in NAME and writes this out to the parameter file using the logical unit number specified in LUN.

SAVEL (LUN, NAME, VARIABLE)

This subroutine encodes the value of a logical variable stored in VARIABLE along with its name in NAME and writes this out to the parameter file using the logical unit number specified in LUN.

SAVER (LUN, NAME, VARIABLE)

This subroutine encodes the value of a real variable stored in VARIABLE along with its name in NAME and writes this out to the parameter file using the logical unit number specified in LUN.

SAVESGNL (ARRAY, MSG, FILENAME, DELAY, OSCFRQ, PMX, MCF)

This subroutine saves the time-domain representation of a signal stored in ARRAY into a data file whose name is contained in FILENAME. DELAY is the time delay associated with the signal. PMX is the percentage of extraordinary polarization in the signal. MCF if the magnetic field correction factor. Then a message contained in MSG is displayed for the user. OSFRQ is the value of the local oscillation frequency that was used in recordeing the signal.

SAVESPEC (ARRAY, MSG, FILENAME, DELAY, FC, FW)

This subroutine saves the frequency-domain representation of a signal stored in ARRAY into a data file whose name is contained in FILENAME. DELAY is the time delay associated with the signal . FC is the central frequency associated with the spectrum, while FW is its associated frequency width. Then a message contained in MSG is displayed for the user.

**SCALE (ARRAY, NDIM, ISTART, NPTS, NC, SCFLAG, EXFLAG, SCF, AMIN,
AMAX, LABEL)**

This subroutine in *ncar.f* or *disp.f* scales NPTS values in ARRAY according to SCFLAG and expands the range of values for plotting purposes depending on EXFLAG. NDIM is the column size of ARRAY and ISTART is the starting index for this array. The scaling factor is returned in SCF. The low and high values determining the range are returned in AMIN and AMAX. The axis label in LABEL is updated with the scale information and a terminating dollar sign.

SERCH (X, XX, N, NVAL)

This routine searches for the index, NVAL, such that XX(NVAL) < X <= XX(NVAL+1).

SETTEC (VALUE)

This subroutine establishes a new value for TEC and other global variables whose values depend on TEC.

SIGDET

This subroutine models the detection of a transionospheric signal by a receiver. It can also generate plots of the received signals.

SIGNAL

This subroutine constructs the input signal and its Fourier transform. It can also generate plots of the input signal.

SORT (N, X, Y, NDIM, NC)

This subroutine sorts the N elements of array X in ascending order and makes sure the the elements in each column of Y are permuted in exactly the same way. NDIM is the column size of Y while NC is the number of columns to permute.

SPCGRM (J, IERSPC)

This subroutine generates a spectrogram for the data values associated with receiver J. If the number of data points to be analyzed is less than the number of points in the spectrogram's time window, or the number of frequency bins is larger than the number allowed in the storage space set aside for the spectrogram, no spectrogram is generated and a non-zero value for IERSPC is returned.

SPECPLOT (OFFSET, ZVAR, ZRANG0, ZSTEP0, ISTT, ISTF, NDT, NDF, NTOTT, NTOTF, NPLOTS, IOPTXY, IOPTZ, SNGL, USRLBL)

This subroutine generates one or more spectrogram plots of the values in ZVAR. The default time values are adjusted by the amount specified in OFFSET. ZRANG0 is the range of values for ZVAL used in generating contours. ZSTEP0 is the step size for contours of ZVAL. ISTT and ISTF are the starting time and frequency indices respectively. NDT and NDF are the number of time and frequency array values used in plots. NTOTT and NTOTF are the time and frequency array sizes respectively. NPLOTS is the number of plots to be generated. IOPTXY specifies the x and y type used in the plots, i.e., linear or logarithmic. IOPTZ specifies the plot option for the z axis, i.e., linear or decibel. SNGL specifies whether a single plot per frame is generated for multiple plot options.

SQRLAW (J, SHIFT, SHFT)

This subroutine passes the signal power through a low-pass filter to determine its envelope. J is the index for a given receiver. SHIFT is a logical variable that when true will calculate SHFT, the receiver delay.

TASKDT

This subroutine performs a Delta Time of Arrival study by simulating the reception of a transionospheric signal by receiver for various values of TEC and estimating the differences in times of arrival.

TASKDU

This subroutine generates various realizations of white noise or scintillations and then simulates the reception of the resulting transionospheric signal by receivers The mean and sigma of the differences in arrival times are calculated.

TASKPS (INDEX)

This subroutine applies one of a number of signal processing algorithms to the transionospheric signal received by the receivers. INDEX in a pointer into the array DTOA.

TASKSD

This subroutine determines what signals are received by each of the receivers whose frequency response has been specified by the user.

TASKSR

This subroutine attempts to reconstruct the original input signal. For each of the nonoverlapping regions where the receivers are active, the FFT of the received signals are divided by the FFT of each receiver response and then by the FFT of the ionospheric transfer function. In other regions the FFT is set to zero. The inverse FFT is then applied to this function.

TASKTP

This subroutine determines what happens to the input signal as it passes through the ionosphere.

TERMSET (TTY, TEKT, IERTTY)

This routine selects the terminal type for interactive execution of the code.

TIMEPLOT (OFFSET, YVAR, IST, ND, NCURVES, NPLOTS, IOPT, SNGL, USRLBL)

This subroutine generates one or more plots in the time domain of the values in the array YVAR such that the values in TIME an array of time values paired in common) are offset by an amount in OFFSET. IST is the starting index of the values to be used while ND is the number of values to be used in the plots. NCURVES is the number of curves to plot on each graph, NPLOTS is the total number of plots, and IOPT is an indicator of the type of plot to make. SNGL is a flag indicating whether single plots per page are requested. USRLBL contains the user specified label, if any, for the plot.

TSHIFT (CFT, TS, SAVE, VAL)

This subroutine multiplies the complex spectrum in CFT by an appropriate function so that the signal in the time domain is shifted by an amount TS. The resulting time domain representation is returned in VAL. SAVE indicates whether the spectrum is to be updated or not.

TVSFNL (J, OFFSET, IERTVSF)

This subroutine analyzes the spectrogram generated for receiver J, to determine the time of peak spectral amplitude for the various frequency bins. These values are plotted and used to estimate the TEC. The default time values are adjusted by the amount specified in OFFSET. If the analysis cannot be performed a non-zero value for IERTVSF is returned.

WIDEBAND (FLO, FHI, ORDER, FLTR)

This subroutine constructs a wideband filter based on a supergaussian function of order ORDER. It passes frequencies between FLO and FHI. The resulting complex function is returned in FLTR.

WINDOW (ARRAY, N, CUTOFF, IST, ND)

This subroutine searches the elements in ARRAY forward from 1 and backwards from N for the first value that exceeds CUTOFF times the largest element in absolute value. The left index is returned in IST while the total number of points in between the limits is returned in ND.

WRITEDT (LUN)

This subroutine formats and uses LUN to write out the values of the parameters associated with a delta time of arrival study.

WRITEDU (LUN)

This subroutine formats and use the LUN to write out the values of the parameters associated with a delta time of arrival uncertainty study.

WRITENDX (NAME, INDEX)

This subroutine formats a string with NAME followed the encoded INDEX, enclosed in parenthesis . For example, NAME (INDEX). Note that there are no spaces between the parenthesis and INDEX.

WRITEPS (LUN)

This subroutine formats and uses LUN to write out the values of the parameters associated with processing signals.

WRITESD (LUN)

This subroutine formats and uses LUN to write out the values of the parameters associated with signal detection.

WRITESR (LUN)

This subroutine formats and uses LUN to write out the values of the parameters associated with signal reconstruction.

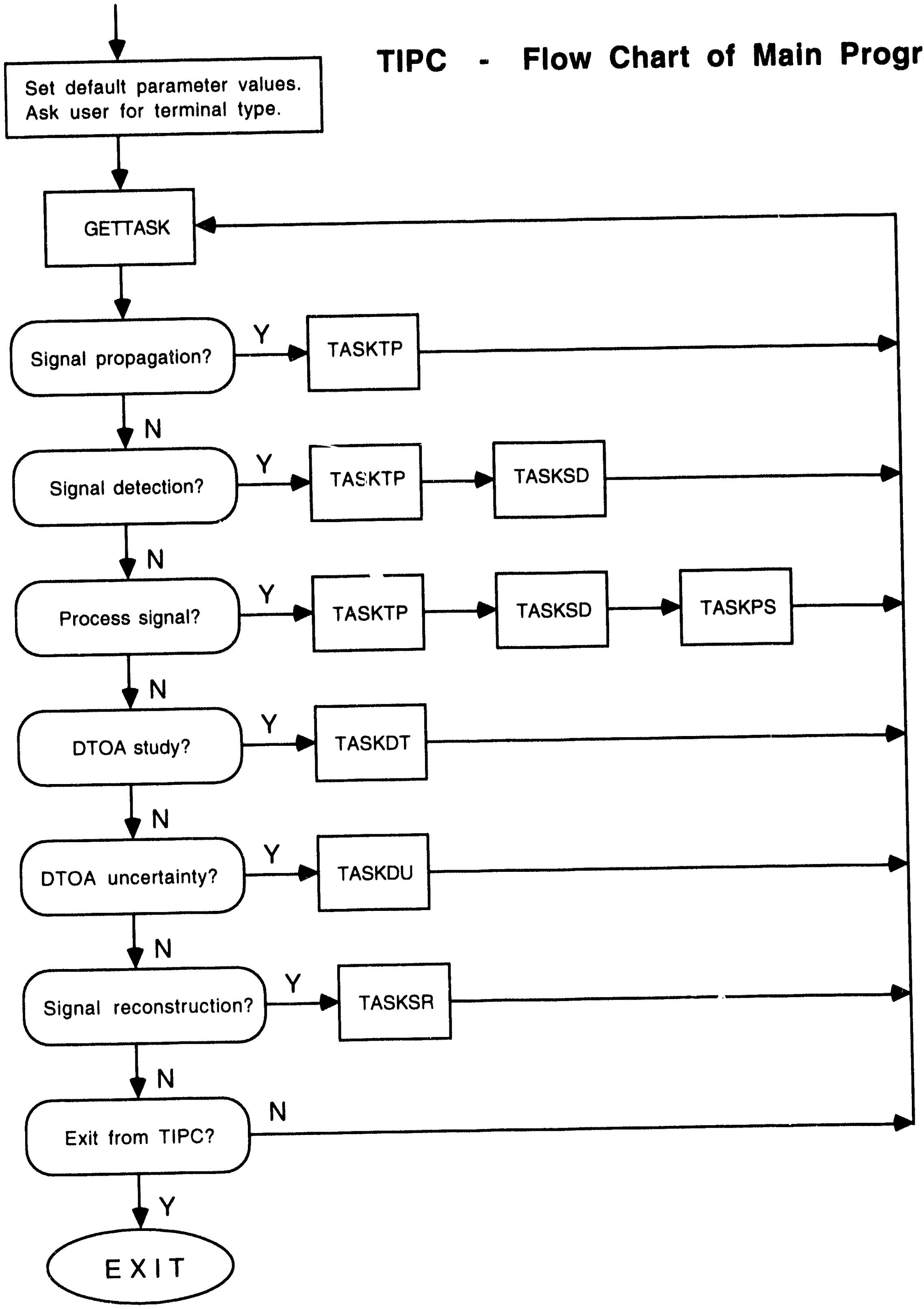
WRITETP (LUN)

This subroutine formats and uses LUN to write out the values of the parameters associated with transionospheric propagation.

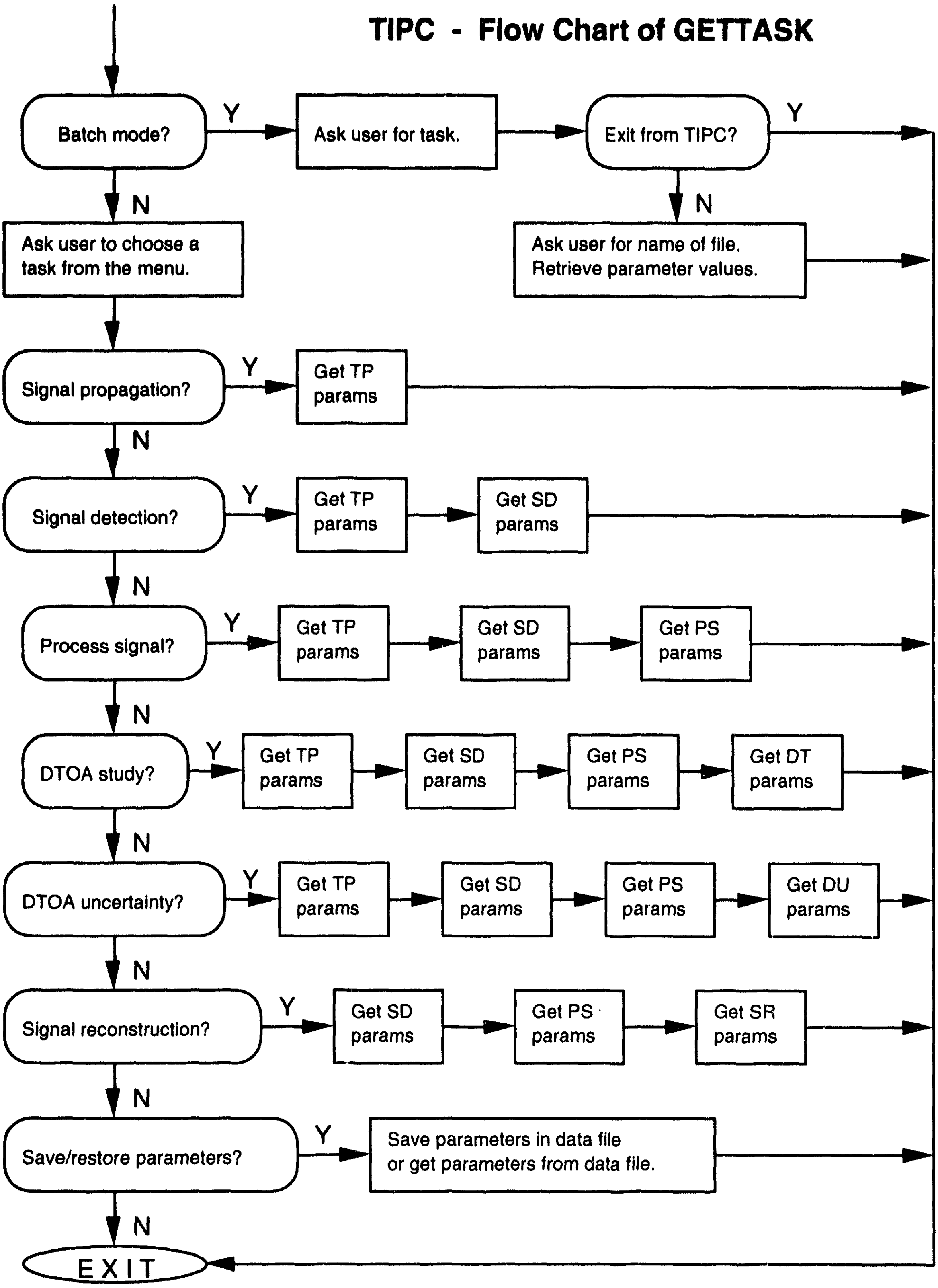
12. Flow Charts

On the following pages there are flow charts which illustrate the overall flow of control in the TIPC software.

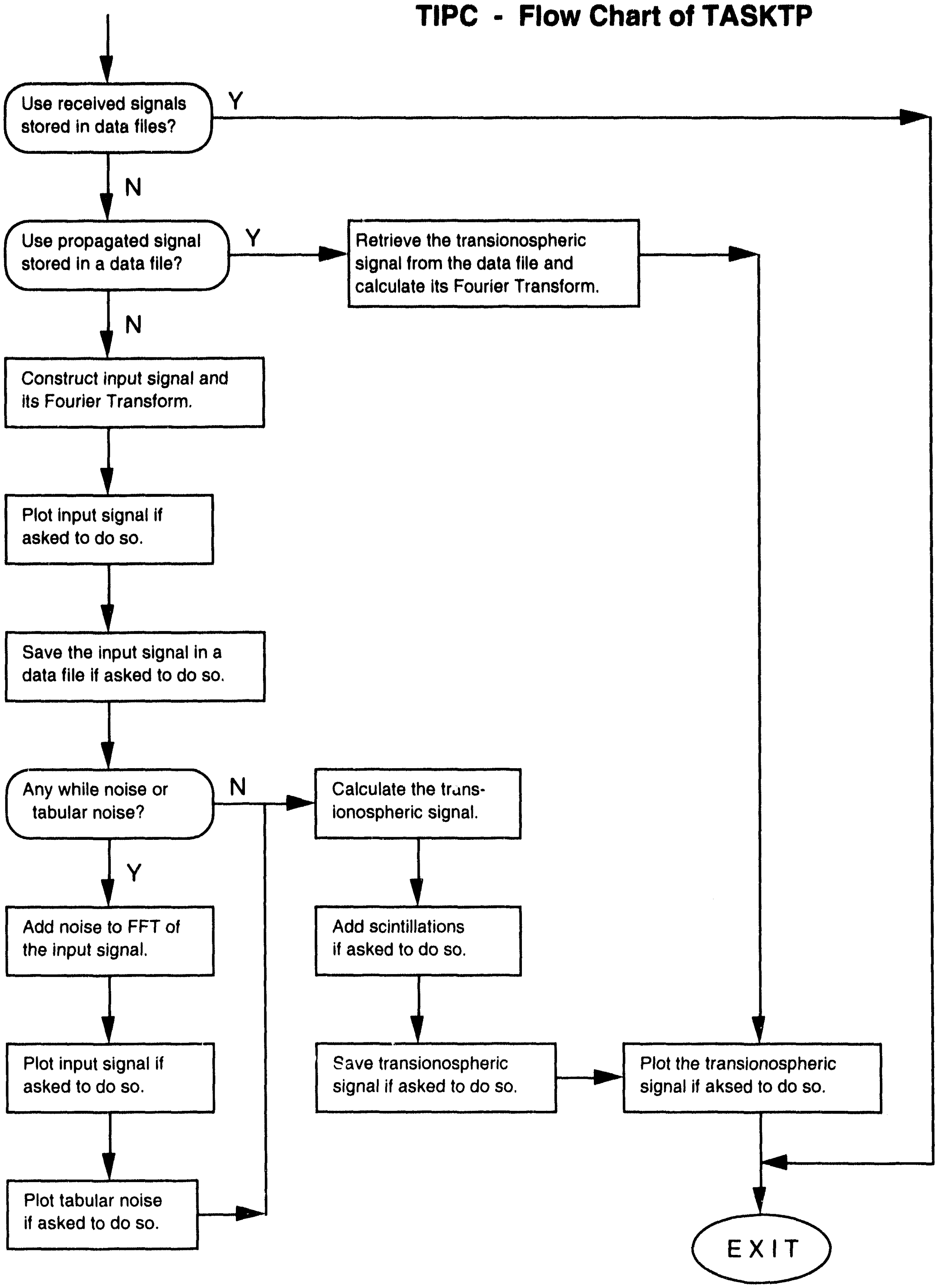
TIPC - Flow Chart of Main Program



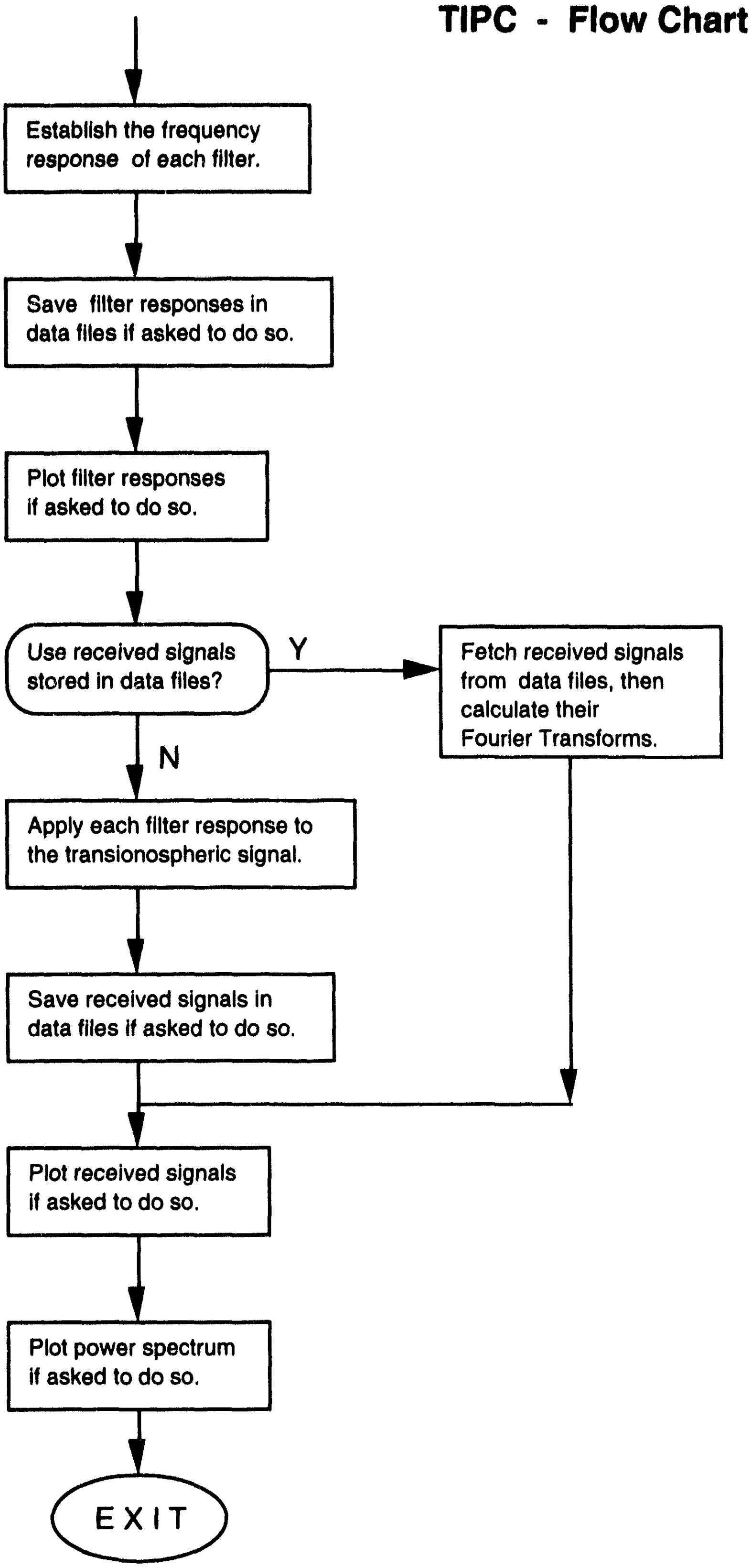
TIPC - Flow Chart of GETTASK



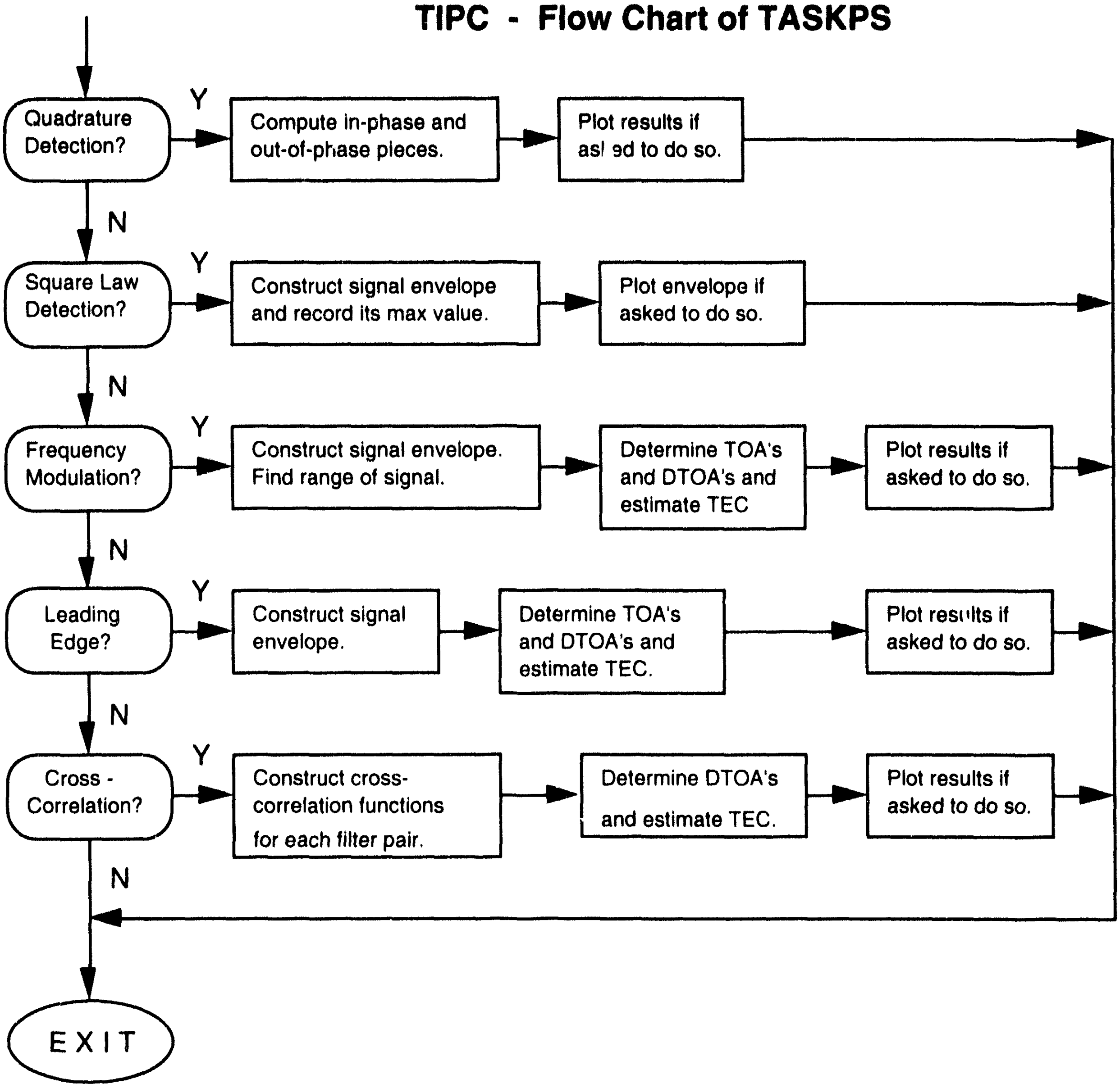
TIPC - Flow Chart of TASKTP


Use received signals stored in data files?
Y
N
Use propagated signal stored in a data file?
Y
Retrieve the transionospheric signal from the data file and calculate its Fourier Transform.
N
Construct input signal and its Fourier Transform.
Plot input signal if asked to do so.
Save the input signal in a data file if asked to do so.
Any while noise or tabular noise?
N
Calculate the trans-ionospheric signal.
Y
Add noise to FFT of the input signal.
Add scintillations if asked to do so.
Plot input signal if asked to do so.
Save transionospheric signal if asked to do so.
Plot the transionospheric signal if aksed to do so.
Plot tabular noise if asked to do so.
EXIT

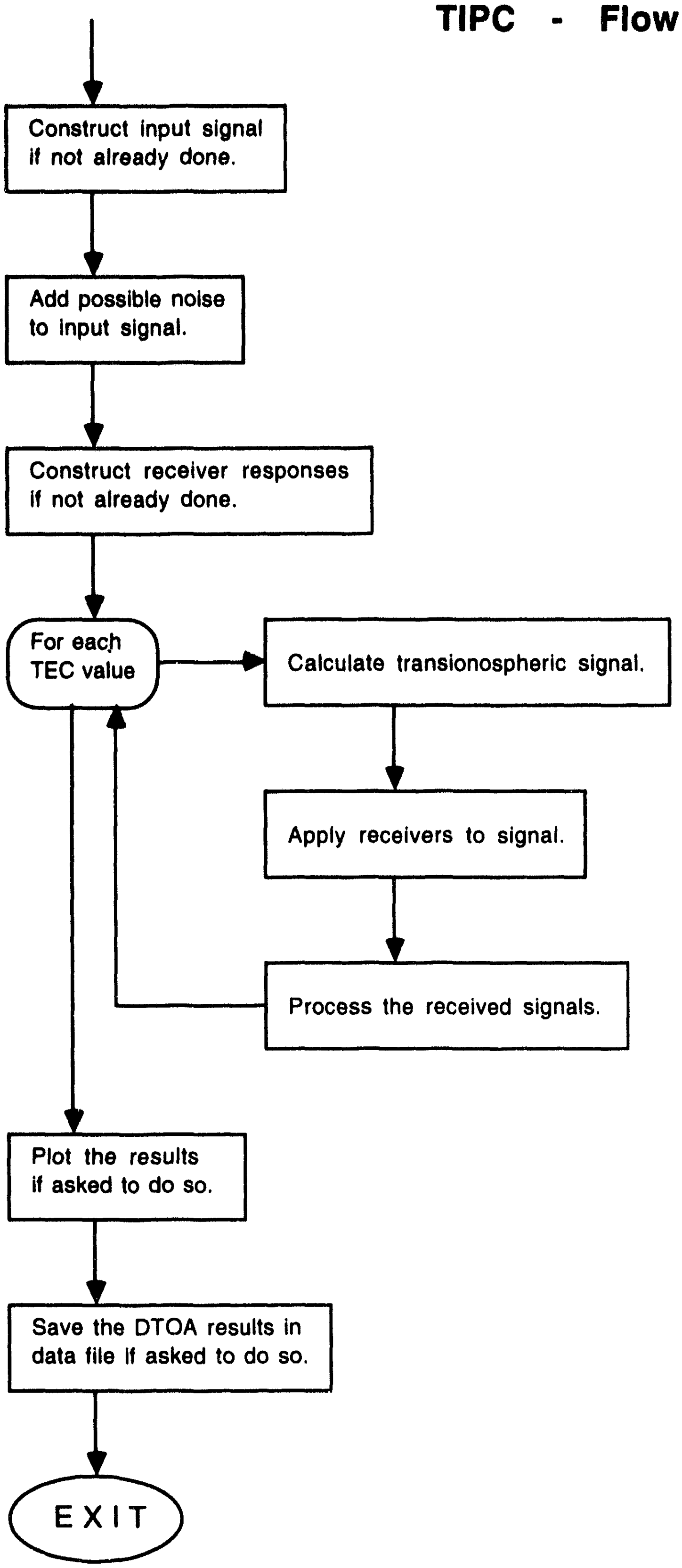
TIPC - Flow Chart of TASKSD


Establish the frequency response of each filter.
Save filter responses in data files if asked to do so.
Plot filter responses if asked to do so.
Use received signals stored in data files?
Y
Fetch received signals from data files, then calculate their Fourier Transforms.
N
Apply each filter response to the transionospheric signal.
Save received signals in data files if asked to do so.
Plot received signals if asked to do so.
Plot power spectrum if asked to do so.
EXIT

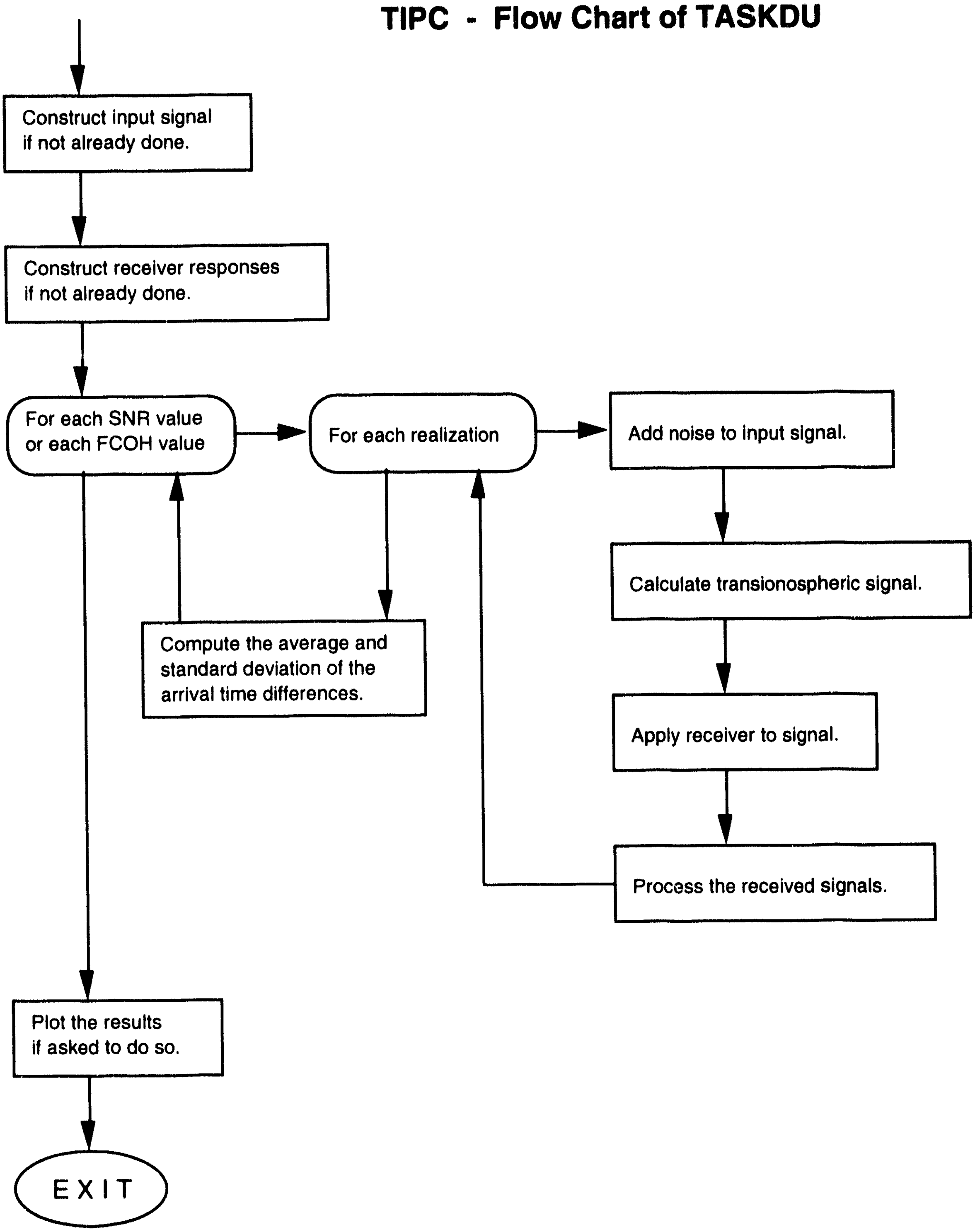
TIPC - Flow Chart of TASKPS


Quadrature Detection?
Y
Compute in-phase and out-of-phase pieces.
Plot results if asl ed to do so.
N
Square Law Detection?
Y
Construct signal envelope and record its max value.
Plot envelope if asked to do so.
N
Frequency Modulation?
Y
Construct signal envelope. Find range of signal.
Determine TOA's and DTOA's and estimate TEC
Plot results if asked to do so.
N
Leading Edge?
Y
Construct signal envelope.
Determine TOA's and DTOA's and estimate TEC.
Plot resr lts if asked to do so.
N
Cross - Correlation?
Y
Construct cross-correlation functions for each filter pair.
Determine DTOA's and estimate TEC.
Plot results if asked to do so.
N
EXIT

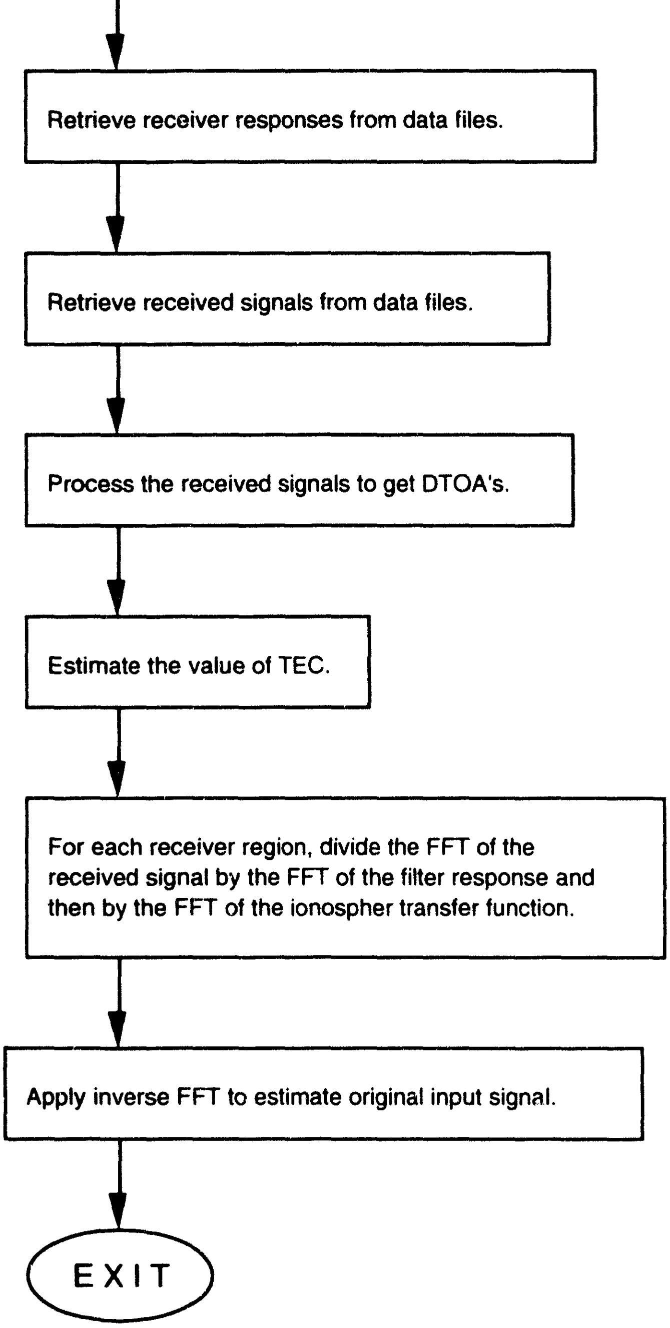
TIPC - Flow Chart of TASKDT


Construct input signal if not already done.
Add possible noise to input signal.
Construct receiver responses if not already done.
For each TEC value
Calculate transionospheric signal.
Apply receivers to signal.
Process the received signals.
Plot the results if asked to do so.
Save the DTOA results in data file if asked to do so.
EXIT

TIPC - Flow Chart of TASKDU


Construct input signal if not already done.
Construct receiver responses if not already done.
For each SNR value or each FCOH value
For each realization
Add noise to input signal.
Calculate transionospheric signal.
Apply receiver to signal.
Process the received signals.
Compute the average and standard deviation of the arrival time differences.
Plot the results if asked to do so.
EXIT

TIPC - Flow Chart of TASKSR


Retrieve receiver responses from data files.
Retrieve received signals from data files.
Process the received signals to get DTOA's.
Estimate the value of TEC.
For each receiver region, divide the FFT of the received signal by the FFT of the filter response and then by the FFT of the ionospher transfer function.
Apply inverse FFT to estimate original input signal.
EXIT

13. References

1. Roussel-Dupré, R. and T.A. Kelley 1990, "*Transionospheric Propagation Code*," Los Alamos Report LA-11799-MS.

2. Oppenheim, A.V. and R.W. Schafer 1989, "*Discrete Time Signal Processing*," Prentice Hall, Englewood Cliffs.

3. Clark, G. L. and L. A. Ankony 1988, "*OINK*", Los Alamos Report CIC # 1125.

4. CLAMS 1988 Los Alamos Report CIC # 148.

14. Figure Captions

Figure 1. The four default input pulse shapes - the delta function representing an impulsive signal; the double exponential representing an electromagnetic pulse (EMP) ; the super gaussian representing a charged particle beam (CPB), and ; the delta hyperbolic tangent also representing a CPB.

Figure 2. The Fourier transforms of the four analytic input pulses shown in Fig. 1.

Figure 3. The amplitude vs. time of the four analytic input pulses shown in Fig. 1 after being propagated through the default ionosphere.

Figure 4. The transionospheric signal profile for the default analytic super gaussian (CPB model) for four values of total electron content (TEC). A TEC value of 1 x 10^{13} e/cm^2 is the default TIPC setting.

Figure 5. The signal amplitude of the default EMP signal with four realizations of white noise added. The power SNR values are 50, 100, 500, and 1000.

Figure 6. The amplitude response of four types of filters. In each case the filter is centered at 150 MHz with a 2 MHz bandwidth.

Figure 7. The spectrogram of a realistic input signal, entitled dish, to which incoherent FM signal has been added.

Figure 8. The tabular pulse *dishnu* which is the actual pulse from a Los Alamos transmitter.

Figure 9. The amplitude of the signal envelope for the two receivers in the input deck *dtoa.prm*. In this case the input pulse is a delta functions, the two filters are butterworth filters (50-100 MHz and 150-200 MHz), and the TEC value is 0.5 x 10^{13} e/cm^2. The time of arrival (TOA), as determined by the leading edge method, is also printed in the lower left portion of each plot.

Figure 10. The delta time of arrival (DTOA) as determined by the leading edge method as a function of TEC. This is the result from the input deck *dtoa.prm* described in the caption of Figure 9. Three TEC values were examined - 0.5, 1.0, and 2.0 x 10^{13} e/cm^2.

Figure 11. The statistics from a very small DTOA uncertainty study. The sample parameter file *delun.prm* provided the input parameters. Two butterworth filters were used. One was centered at 50 MHz and the second was centered at 150 MHz. Each had a 2 MHz bandwidth. Three realizations of white noise were added to the default EMP signal at two SNR values - 100 and 800.

* U.S. GOVERNMENT PRINTING OFFICE 1993-0-573-024/87049