The HIFT Data Set

The University of Michigan HIFT Group

June 4, 2002

Contents

1 Introduction

The Heard Island Feasibility Test (HIFT) was a multinational project to investigate the feasibility of studying the propagation of sound in the ocean on a global scale. HIFT was the creation of Professor Walter Munk of the Scripps Institution of Oceanography.

A large number of people contributed to the HIFT effort. This set of CDs archives the data that was acquired and makes it freely available for study by all who are interested. We offer our apologies for taking so long in the creation and distribution.

For information about the Heard Island Feasibility Test see: *The Journal of the Acoustical Society of America*, Vol. 96, No. 4, October 1994, pp 2327-2484..

Not included in this CD set are:

- Data from DREA in Canada. Not received.
- Data from DREP in Canada. Not received.
- Data from WHOI. Problems (self-inflicted at the UofMich) were encountered reading the data tapes. A replacement copy was requested but not received.

Depending on the site, a HIFT data set can contain data from 1 up to 8 A/D channels and is nominally 80 minutes in duration. The sample rate was 228 Hz.

A significant amount of effort was expended to allow verifying the correctness of the data files both before and after copying from a CD into a computer system.

2 The CD set

The HIFT CD set consists of 6 disks.

CD 1 contains

- PDF files containing survey plots of the data. These are intended to provide a visual guide into the data.
- the GPS records acquired at the source site during the signal transmissions.
- MATLAB code that can be used to access and process the data. These scripts illustrate how MATLAB can be used to access the data.
- the source code for programs to generate and check cyclic redundant checksums (CRCs).
- miscellaneous items.

CDs 2 through 6 contain the raw data sets.

Data files use the first letter in their name to identify the data source. In the case of the Ascension Island data there were two data acquisition computers. The digit following the initial a identifies computers 0 and 1. Otherwise the remainder of the file name consists of a three (two for Ascension) digit year day followed by hour and minute corresponding to the start of data acquisition. The Russian data sets (the former Soviet Union) data sets were received split out into "seances" and this nomenclature was preserved when they were refomatted to be compatible with the HIFT data format. Information is included in the description of the Russian data set contained below that relates the seances to the time of data acquisition.

The six CD set was mastered using a PC running Microsoft Windows 98. The root directory of each CD contains a list of files and a corresponding set of cyclic redundant checksums (CRCs). The `crctest` program can be used to compute the CRCs of the file contained on a given CD and compare with them against the values in this list.

The CD masters along with the `crctest` program were tested on an NT system, a Windows 98 system and on an HP UNIX system.

The Windows NT/W98 executable of `crctest` is included in the root directory of CD 1. The source is also included on CD 1 in the directory `\0code\crctest`. The program can be compiled for UNIX systems and has been tested on an HP UNIX system. Problems may or may crop up when using other UNIX systems. The source code for `crctest` should be modified as necessary. The program is written in C and was tested on the HP using the GNU C compiler `gcc`.

The data was written on the CDs using the ISO-9660 CD format. Files copied from a CD to a hard disk generally inherit read-only status. The access permissions must be changed when it is desired to modify the contents of a file (i.e., when modifying a MATLAB code example).

2.1 CRC usage

A cyclic redundant checksum (CRC) was computed for each data set containing sample values contained on this CD set. The CRCs are meant to provide a means of checking the integrity of the data files contained on the CDs and also after copying them to a processing system. The CRCs were generated after the data files were read from the original data tapes and prior writing them onto the CD masters.

It is unlikely that there will be any problems with copying the data sets from the CDs to processing computers but rare events do occur and thus a means is provided to verify that the data files are as expected.

The CRC generation program used to generate the original on-site CRCs is contained on CD 1 in the `0code` subdirectory under the name `atoccrc.c`. This program was originally written for use by the ATOC project. The CRC generator was initialized at zero. This was not the best choice because the resulting CRC will not check for varying lengths of zero values at the beginning of a file and all files containing only zero values produce the same CRC value. Another problem with `atoccrc` is its minimal user interface essentially requiring manual generation of file lists.

A second CRC generation/checking program was written `crctest` for use in generating CRCs for the HIFT CD data set. The generator is initialized with all ones and the user interface is more useful. The program has a mode that allows checking the older CRC values using the file lists contained on the CDs.

The program `crctest` was originally written as a Windows console application. It may be executed using the DOS box prompt or interactively. The program was ported to an HP UNIX system. The user interaction works well with the UNIX environment. The program was also minimally tested on a SOLARIS system.

The files containing the CRC values associated with each data set are contained on CD 1 in subdirectories under the directories containing the spectra PDF files.

2.1.1 atoccrc

This program is included only because it was used to generate the on-site CRCs.

2.1.2 crctest

The `crctest` program replaces the earlier `atoccrc` program and provides more user friendly support for generating and checking CRCs. The program can be compiled for use on either a Windows or a UNIX platform. When compiled for use in UNIX the program converts the backslash (\) characters used in the CRC file lists into forward slash (/) characters.

The `crctest` program can be run at the command line or via a batch file using the following arrangement of arguments:

`crctest` Enters interactive mode.

`crctest crcfile` Assumes that the files involved are located in the current directory. When generating CRCs the CRC values and the file names are written to `crcfile`. When verifying, `crcfile` is assumed to contain a list of files and their associated CRC values. The named files have their CRC values generated and compared against the given values. Differences in values cause error information to be printed onto the console.

`crctest start_directory crcfile` The file names contained in the `crcfile` start with `.\`. When verifying the `.\` is replaced by the named starting directory (including drive). This feature essentially allows the path to the data files to *float*.

Options are:

`-h` Used to verify the original `atoccrc` generated CRC files.

-H Only changes the initial CRC value from all ones to all zeros. This provides compatibility with earlier CRC calculations.

-g Generate CRC values and write them into the designated CRC fille.

-l Causes the program the program to convert file path/names read from the CRC file to lower case. Depending on the mode of mounting this was useful on the HP UNIX system.

-u Causes the program to convert file path/names read from the CRC file to upper case. Depending on the mode of mounting this was useful on the HP UNIX system.

The command line is first scanned for options. These are noted and then removed. The remaining input list is then parsed described as above.

Example:

Assume that HIFT_CD04 is mounted on drive `h:`. The following command can be used to check all files (except the crc file itself) contained on that CD.

```
crctest   h:\   h:\cd04crc.txt
```

The equivalent run on the test HP UNIX system would use the command line:

```
crctest   /cdrom   /cdrom/cd04crc.txt   -l
```

Only files having CRC mismatches cause message lines to be printed. When the program completes execution it provides a count of the directories and files that it has accessed.

2.2 CD 1

CD 1 contains the summary spectra plots for all data sets. The plots are contained in PDF files and can be viewed and/or printed using Adobe's Acrobat Reader. Section 3 discusses these plots in more detail.

There may be a problem when printing these PDF files using a Hewlett/Packard PCL printer. The dots used to make up the spectra plot likely will not print using Acrobat Reader using its default configuration. They can be printed on such a printer using Acrobat Reader 4.0 by placing a check mark in the box titled *Print as image* located at the top right side of the print menu. No printing problems have been experienced when using Ghostscript.

CD 1 contains approximately 260 Mbytes.

```
\                   Root directory
\0Code\             Used to hold HIFT related code
\0Code\atoccrc\     The older CRC program
\0Code\crctest\     The "current" CRC program
\0Code\MAD\         Code supplied by Matt Dziecicuh
\0Code\MAD\gps\     Shipboard GPS records
\0Code\MAD\mseq\    Useful \MATLAB\ code
\0Code\MATLAB\      \MATLAB\ support to access HIFT data
\0Code\misc\        Other software relative to HIFT
\0Code\rusbin\      Code to convert Russian data format
\0Code\spectra\     Code for the spectra program
```

The remaining directories contain the survey plot PDF files.

```
\A1\          Ascension Island computers 1 and 2
\A1\MISC\
\A2\          Ascension Island digitized analog tapes
\A2\MISC\
\B1\          Bermuda data part 1
\B1\MISC\
\B2\          Bermuda data part 2
\B2\MISC\
\C\           Capetown
\C\MISC\
\G\           Goa
\G\MISC\
\H\           Heard Island
\H\MISC\
\K\           Kerguelen
\K\MISC\
\M\           Mawson Station
\M\MISC\
\S\           Russia (Soviet Union)
\S\MISC\
\T\           Tasmania
\T\MISC\
\W\           Western United States
\W\MISC\
\X\           Christmas (XMAS) Island
\X\MISC\
\Z\           New Zealand
\Z\MISC\
```

2.3 CD 2

CD 2 contains approximately 611 Mbytes.

```
\
\A1\     Ascension Island computers 1 and 2
```

2.4 CD 3

CD 3 contains approximately 610 Mbytes.

```
\
\A2\     Ascension Island digitized analog tape
\C\      Capetown
\G\      Goa
```

2.5 CD 4

CD 4 contains approximately 509 Mbytes.

```
\
\B1\     Bermuda data part 1
\H\      Heard Island
\K\      Kerguelen
\M\      Mawson Station
```

2.6 CD 5

CD 5 contains approximately 620 Mbytes.

```
\
\B2\     Bermuda data part 2
\S\      Russia - Krylov Seamount
```

2.7 CD 6

CD 6 contains approximately 633 Mbytes.

```
\
\T\     Tasmania
\W\     Western United States
\X\     Christmas Island
\Z\     New Zealand
```

3 Reception survey plots

The survey plot generation process is described on page 2347 of the JASA issue; an example is shown in Figure 1.

Each channel of each data set was processed and the results plotted. The plots were converted into PDF form and are located on CD 1. These plots provide a ready visualization of the quality of each channel of reception. The plots are easily view using Adobe's Acrobat Reader (or a similar program) and can be readily converted into hard copy form.

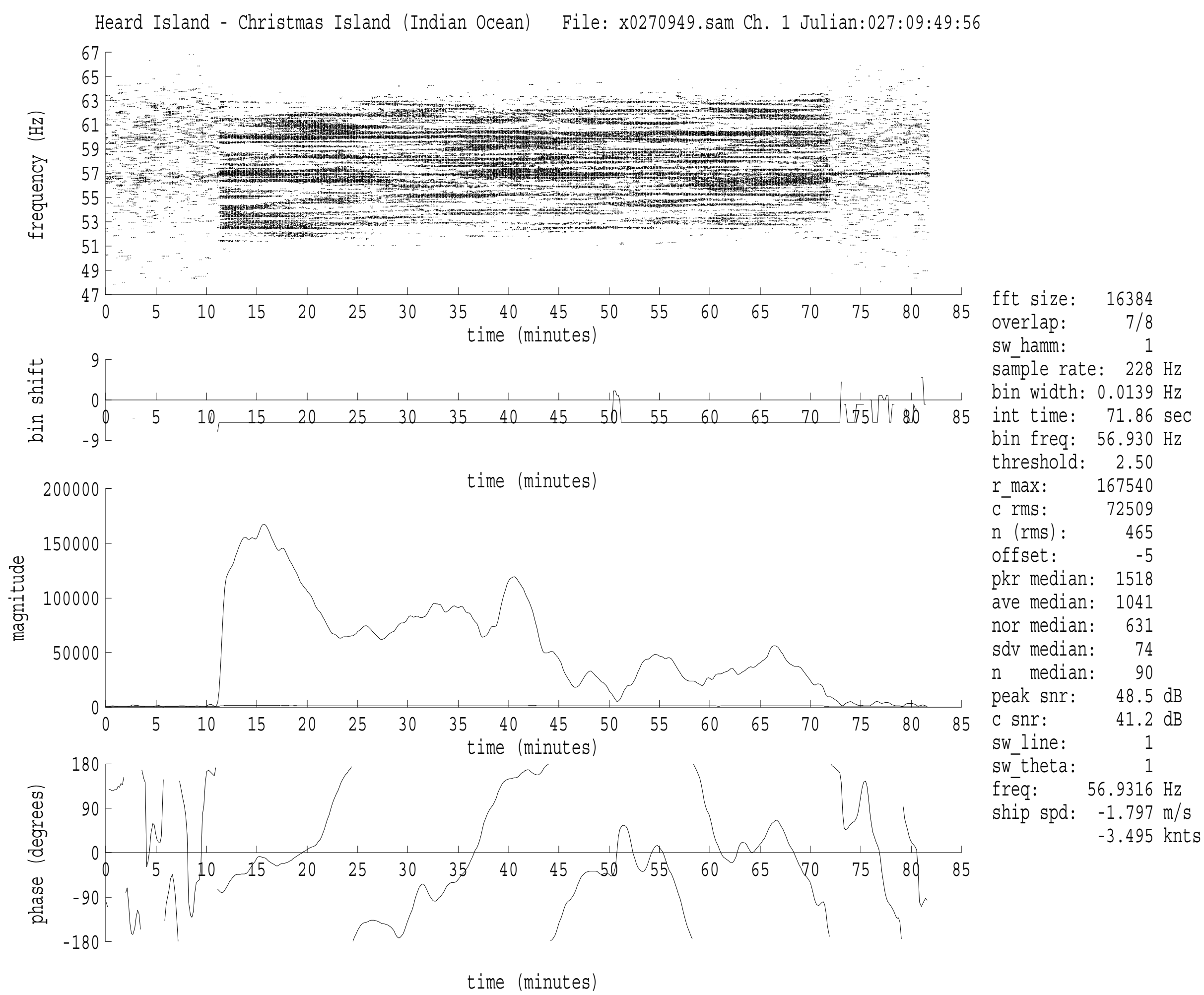

Figure 1: Survey plot obtained using data set x0270949.sam.

The survey plots provide information about the quality of the data as well as numerical data such as signal-to-noise ratio and ship speed. Correctly demodulating the reception and correcting for Doppler requires a "v/c" factor; v is the estimated projected speed in m/s determined for the survey and listed in Section

6; *c* is the path average speed of sound estimated as 1498 m/s.

The survey plots also show various receiver problems.

Ascension Island had a problem termed *a non-linear ground fault* which was corrected between the A0271030 and A0271330 receptions. We believe that the receptions prior to January 27 at 1330 hours are fatally corrupted. The survey evidence is the three lines at 57, 60, and 63 Hz shown on CW transmission A0261330 and A0270130 (which should have shown only a 57 Hz line), and similar indications on all of the receptions before A0271330.

Most ship stations show a repetitive interference pattern. The actual receiving buoy radioed its signal to its ship; the ship tried to stay near enough to hear the radio signal but far enough away to reduce its acoustic jamming of the buoy. As the ship sailed back and forth its jamming rose and fell. We call this *own ship* noise.

The strong signal at X=Christmas Island, at a 5510 km range, makes X a good source of high-to-signal noise survey plots. For example, X0270049, X0270349, and X0270649 show similar patterns of *own ship* noise at about 51 and 59 Hz on CW, Pentaline, and M1-sequence transmissions respectively. However, X0270494 does not show *own ship* noise.

The effect of median-plot-normalization is evident in most spectrogram plots (the top survey plot), and could lead to misinterpretation. The plot time step is about 9 seconds. At each step every spectral magnitude that is 2.5 times the median spectral amplitude is shown as a dot. When *own ship* noise increases, the median increases and it appears that the signal becomes weaker; in truth, the signal becomes relatively weaker, not absolutely weaker.

4 Working with the HIFT data using MATLAB

A basic suite of MATLAB scripts and functions are included with the data for use in accessing and working with the HIFT raw data files.

The plain text (ASCII) MATLAB files are contained on CD 1 in the directory \0CODE\MATLAB. On a Windows system these remain read-only when copied onto a hard disk. If it is desired to modify the code it is necessary to either set the file attributes to allow writing or to write the modified code into a write enabled file.

These m-files are included to illustrate how to access and work with the HIFT data. They can be modified and used as desired. For example the output from `hiftdemo.m` retains only the magnitude of the multipath reception in order to generate Figure 2. To obtain a complex valued output requires only removing one `abs(  )` from one MATLAB line.

`hreadchn`: A function used to read HIFT data into a MATLAB program.

hdedop: A function used to demodulate and resample to correct for the effects of fixed Doppler compression/expansion.

The data is first shifted in frequency by multiplying by the complex exponential (Fourier frequency shifting theorem) and then linearly interpolated to resampled in order to remove the effects of constant Doppler. In order to provide researchers flexibility, lowpass filtering is not performed.

fmst: This function implements the fast m-sequence transform effectively removing the energy time spreading that results from using m-sequence modulation.

fmstset: This function is used to setup the m-sequence specific information needed by fmst.m.

hdemo: A MATLAB script used to compute time sequential multipath responses from a HIFT sample file. The magnitudes are saved in an file for later processing or plotting.

A simple sliding sum filter ("sinc") is used to filter the demodulates generated by hdedop. The filtering is cyclic to the sequence period.

hiftwfp: A MATLAB script to generate a time windowed waterfall plot of the multipath structure for a full reception.

Matthew Dzieciuch at Scripps has made available his MATLAB programs for working with digitized m-sequence based waveforms. These were written for use with data from a later experiment, ATE, and are intended to illustrate how to work with m-sequence based data.

4.1 hreadchn.m

This script extracts the time series associated with a specified data channel. Setting the third calling argument non-zero causes the ASCII file header to be displayed followed by an information line indicating the number of values read.

```
function [sam, time_text] = hreadchn(samfile, channel, details)
% function [sam, time_text] = hreadchn(samfile, channel, details)
%
% reads HIFT sample file and extracts specified channel's
% time series
%
% samfile    is text string containing name of the input file
% channel    is the number of the channel to extract
% details    if non-zero prints header and other information
%
```

```
% sam        returned time series ... column
% time_text  string containing Julian time string from header
%
% KM 10Aug2000 based on code supplied by Matt Dzieciuch

fid=fopen(samfile, 'r', 'ieee-le'); % open file little endian

% read header (set details to non-zero value to print header)

N_chans=0;
chan_index = 0;
line=fgetl(fid);
if details disp(line); end
while strcmp(line, '*DATA*') == 0
  line=fgetl(fid);
  if details disp(line); end
  off=findstr(line, 'Julian:');
  if off
    time_text = line;
  end
  off=findstr(line, 'channel:');
  if off
    N_chans = N_chans+1;
    chan_num = str2num(line(off+8:off+9));
    if chan_num == channel
      chan_index = N_chans;
    end
  end
end

if chan_index == 0
  error(sprintf('channel %d not found in input file\n', channel));
end

start=ftell(fid); fseek(fid,0,'eof'); stop=ftell(fid);
fseek(fid,start,'bof'); sizs=(stop-start)/2;

scan_count = floor(sizs/N_chans);

if N_chans == 1
  [sam count] = fread(fid, sizs, 'int16');
else
  [temp count] = fread(fid, chan_index-1, 'int16');
  % try to read extra scans and see if it does...paranoia
  [sam count] = fread(fid, scan_count+100, 'int16', 2*(N_chans-1));
end

if count ~= scan_count
  error(sprintf('computed scans: %d  read: %d\n', scan_count, count));
end
```

```
fclose( fid);

if details
  fprintf('size: %d  chans: %d  scans: %d  channel: %d  chan_index: %d\n', ...
    sizs, N_chans, scan_count, channel, chan_index);
end

return
```

4.2 hdedop.m

This script frequency shifts a data set from a Doppler'ed carrier to base band. The starting values are real valued and be come complex value in the downshifting process. Both the original positive and negative frequency components are shifted. The shifting is accomplished by directly applying the Fourier theory shifting theorem.

The sample rate used at the receivers was for zero Doppler. The time spread removal processing that follows requires a precise integer number of samples per modulation bit. The presence of Doppler requires adjusting the data sample times using a interpolation process. For hdedop linear interpolation is used.

The returned downshifted, resampled data set is NOT filtered. It remains to filter the returned values to remove the negative frequency component now nominally at minus twice the carrier.

```
function [demods, Eflag] = hdedop(samples, VonC, FConFS, start_scan, n_scans)
% Function to frequency shift and adjust sample times to account for
% Doppler compression/expansion when processing HIFT data sets.
%
% samples      Column vector of sample values extracted from a "standard"
%              HIFT data set.
%
% VonC         Ratio of estimated source velocity as seen at the site that
%              generated the sample data set divided by the estimated speed
%              of sound.
%
% FConFS       The ratio of the carrier frequency (FC) divided by the sample
%              rate (FS).
%
% start_scan   The scan to start the output at.  This is relative to the
%              start of the data set AFTER doppler correction.  For example
%              if there are N samples per period of reception.  The starting
%              indices of successive periods are 1, N, 2N, 3N, ... .
%
% n_scans      The number of dedoppered values to generate
%
% demods       A column vector of complex downshifted, resampled to compensate
```

```
%               for doppler.  These have not been filtered!!!  There remains
%               the -2fc term.
%
% Eflag         This is zero if the requested number of values has been
%               generated.  Otherwise it is nonzero.
%
%               22April2000 initial production version...K.Metzger

Eflag = 0;

d_start_scan = (start_scan-1)/(1+VonC);        % start scan in doppler data
d_end_scan = (start_scan+n_scans-1)/(1+VonC);  % end scan in doppler data

d_start = floor(d_start_scan);
d_end = floor(d_end_scan)+1;
d_N = d_end-d_start+1;

if d_end >= length(samples)
  demods = [];
  Eflag = 1;
  return;
end

demods = exp(-2i*pi*(1+VonC)*[d_start:d_end]*FConFS).'.*samples([d_start:d_end]+1);

d_index = [start_scan-1:start_scan-1+n_scans-1]'/(1+VonC)-d_start;

temp = demods(floor(d_index)+1);
demods = temp + ...
  (d_index-floor(d_index)).*(demods(floor(d_index+1)+1)-temp)*(1+VonC);

return;
```

4.3 fmst.m and fmstset.m

One way to make multipath measurements at a given frequency is to use a short carrier burst. The shorter the burst the better the time resolution of multipath structure. The minimum useful duration of the burst is limited by the bandwidth of the acoustic source. Typically the bandwidth of the burst is to be reasonably well matched to the bandwidth of the source. Making the burst bandwidth much wider that the source's does not significantly increase the time resolution. Making the burst bandwidth significantly narrower than the source's resolution reduces the time resolution.

The received level can be built up by transmitting identical time separated bursts (digits) and coherently summing the resulting receptions. The time separation between digits needs to be precisely controlled and should exceed the expected time spread. The number of responses that can usefully be coherently averaged is determined by changes in propagation, source movement, and transmitter/receiver timing stability.

The effective received level can also be increased by use of a pulse compression waveform (PCW). A good PCW will place energy uniformly over the interval used above by successive digits. The energy contained in the transmitted waveform is increased by the number of digits that can be fit between the above isolated digits. Call this L. The received waveform can be processed in a way that in effect collapses the received waveform into a waveform equivalent to using repetitive isolated digits but having L times the amplitude. This processing also has an effect on the amount of noise with the result being an increase by a factor of L in the signal-to-noise ratio of the PCW based reception as compared with the repeated isolated digits reception.

The HIFT waveforms used periodic binary linear maximal sequences to phase modulate the carrier. The modulation angle used by HIFT was ± 45 degrees.

The processing implemented by the two scripts in this section collapses the energy contained in the L digits is collapsed into a single digit duration but differs from matched filtering in that it does not implement the match to the digit wave shape.

The `fmst` implements the time spread removal using the Hadamard transform. Relevant references are listed in the comments contained at the beginning of the MATLABsource.

Important restrictions are that the data set being processed corresponds to one period of the modulation and that there be precisely an integer number of samples per modulation digit. The *single period* may also be the coherent sum of several periods.

```
function out = fmst(in, p, angle)
%Function to implement fast m-sequence transform.
%
% out = fmst(in, p, angle)
%
% in      a column vector or a matrix whose columns consist of
%         samples of one period of a m-sequence based baseband
%         waveform.  The number of samples corresponding to
%         a sequence bit must be an integer value.
%
% p       a structure generated by the fmtsetup function.  This
%         is sequence law dependent and provides the fmt
%         function information needed specific to the sequence.
%
% angle   modulation angle used in generating the input data
%         set.  This is in degrees.  Values greater than 180
%         result in the use of the "matched" angle,
%         angle=atan(sqrt(L)) where L is the number of bits
%         in the sequence period.
%
```

```
%         The basic m-sequence transform technique is described
%         in the paper "On Fast M-Sequence Transforms" by
%         M.Cohn and A.Lempel, IEEE Transactions on Information
%         Theory, January 1977.
%
%         Hadamard processing based on T.G.Birdsall's pass invariant
%         fast Hardamard algorithm.  See also "Algorithms for
%         Discrete Fourier Transform and Convolution, 2nd Ed."
%         by R.Tolimieri, M.An, and C.Lu, Springer-Verlag, 1997.
%         Chapter 2 discusses tensor product relationships.
%
%         22April2000 initial production version...K.Metzger

layers = p.layers;          % layers in transform
LL = 2^layers;              % sequence length plus 1
L = LL-1;                   % sequence length
LL2 = LL/2;
[rows, cols] = size(in);    % determine dimensions of input

if angle>180                % if > 180 degrees use matched angle
  ta = sqrt(L);
else
  ta = tan(pi*angle/180);   % otherwise use specified angle
end

% compute bias (aka pedestal) correction factor

factor = (ta*ta-L + ta*(L+1)*j)/(L*L+ta*ta);

out =zeros(rows,cols);      % create output using desired size
for c=1:cols                % loop on the number of input columns

  % extract a column from the input
  % place a zero at the column top
  % reshape to obtain one sample per bit and a number of
  %   columns equal to the number of samples per bit
  % scramble values to allow use of Hadamard processing

  work(p.in,:) = [zeros(1,rows/L); reshape(in(:,c), rows/L,L).'];
  %tic;

  % loop on the number of tensor products needed to form the
  %   Hadamard matrix using the 2x2 version

  for lay_ctr = 1:layers

    % the algorithm is layer invariant and does all
    %   sample per bit interleaves simultaneously
    %
    % work = [work(1:2:L,:)+work(2:2:LL,:); work(1:2:L,:)-work(2:2:LL,:)];
    % but runs faster if done in stages
```

```
    tempa = work(1:2:L,:);
    tempb = work(2:2:LL,:);
    work(1:LL2,:)    = tempa+tempb;
    work(LL2+1:LL,:) = tempa-tempb;

    % processing can also be implemented as
    %
    % work([1:2:L 2:2:LL],:) = ...
    % [work(1:LL/2,:)+work(LL/2+1:LL,:); work(1:LL/2,:)-work(LL/2+1:LL,:)];
    % but runs faster if done in stages

    %tempa = work(1:LL2,:);
    %tempb = work(LL2+1:LL,:);
    %work(1:2:L,:)  = tempa+tempb;
    %work(2:2:LL,:) = tempa-tempb;

  end
  %toc

  % correct the bias (dc offset) for each column based on each
  %   column's sum unscrambling the bit order at the same time

  work = work(p.out,:)-kron(ones(LL,1), work(1,:))*factor;

  % place the result into the output array doing a reinterleaving
  %   of samples

  out(:,c) = reshape(work(2:LL,:).', rows,1);

end

return;
```

The `fmstset` script is to be called prior to use of `fmst`. This script sets up the permutation matrices to be used as part of the Hadamard processing.

```
function perm = fmstset(law)
% Function to generate setup information for fast m-sequence
% transform (fmst) function.
%
% law     sequence law written as if it were octal.  For to
%         specify the law 2033 written in octal use the value
%         2033 decimal.
%
% perm    A structure containing the number of transform layers
%         and two arrays of permuation vectors.  The vectors
%         are used by fmt to unscramble and rescample indices
%         so that processing can be performed using the fast
%         Hadamard transform.
%
```

```
%         The basic m-sequence transform technique is described
%         in the paper "On Fast M-Sequence Transforms" by
%         M.Cohn and A.Lempel, IEEE Transactions on Information
%         Theory, January 1977.
%
%         22May2000 initial production version...K.Metzger

% convert law written in decimal to binary

value = 0; weight = 1; temp = law;
while temp ~= 0
  value = value+weight*rem(temp,10);
  temp = floor(temp/10);
  weight = weight*8;
end

% determine the number of transform layers
%   and the period length, L

layers = 0; v = floor(value/2);
while v ~= 0
  layers = layers+1;
  v = floor(v/2);
end

L = 2^layers-1;
end_bit = (L+1)/2;
ms_c = 1;
ss_c = 1;
scrm = zeros(1,L+1);
unsc = zeros(1,L+1);
scrm(1) = 1;
unsc(1) = 1;

% generate the permutaion index arrays

for idx = 2:L+1
  scrm(idx) = ss_c+1;
  unsc(idx) = ms_c+1;
  temp = bitand(ss_c, value);
  ss_c = floor(ss_c/2);
  while temp ~= 0
    if bitand(temp, 1)
      ss_c = bitxor(ss_c, end_bit);
    end
    temp = floor(temp/2);
  end
  if bitand(ms_c, 1)
    ms_c = bitxor(ms_c, value);
  end
  ms_c = floor(ms_c/2);
```

```
end

perm.layers = layers;  % number of transform layers to be used
perm.in = scrm;        % permutation to be applied to input
perm.out = unsc;       % permutation used to order the output

return;
```

4.4 Test and demonstration scripts

4.4.1 hdemo.m

This script was used to generate the values plotted in Figure 2.

It took approximately 23.5 seconds to process the 80 minutes of single channel data contained in the file X0270949.sam using this script and a 1 GHz Pentium 4 based PC.

```
% Script to dmonstrate processing HIFT data using MATLAB
%
% 5June2000 initial version...K.Metzger

clear all;
tic;                      % start timer running

SoundSpeed = 1498;        % meters per second
details = 1;              % turn on input log output

fname = 'x0270949.sam';   % process this data file
signal = 'M2';            % waveform M2 was sent
VonC = -1.798/SoundSpeed; % Doppler estimate from earlier processing

channel = 0;              % process this channel (names start at 0)

% read the desired channel's sample values
%
samples = hreadchn(fname, channel, details);

% set up the sequence law based on transmitted waveform
%
if signal == 'M1'
  law = 537;
elseif signal == 'M2'
  law = 1473;
elseif  signal == 'M3'
  law = 2033;
elseif signal == 'M4'
  law = 5747;
else
  error('invalid signal designator\n');
end
```

```
FConFS = 57/228;   % carrier to sample rate ratio
sam_cyc = 4;       % samples per carrier cycle
cyc_dig = 5;       % number of carrier cycles per sequence bit
angle = 45;        % modulation angle
Qfilter = 4;       % number of samples in periodic sliding filter summation

perm = fmstset(law);      % set up permutation information for fmt
L = 2^perm.layers-1;        % number of sequence bits per period
sam_dig = sam_cyc*cyc_dig;  % samples per sequence bit (digit)

start_scan = 1;       % starting A/D scan to use in processing
n_scans = L*sam_dig;  % number of A/D scans per sequence period
Eflag = 0;            % clear the data flag
PeriodCount = 0;      % initialize the number of sequence periods read

ns = n_scans/2;              % this mimics earlier processing start point
start_scan = start_scan+ns;  % by skipping 1/2 half period worth of A/D scans

while Eflag == 0             % read periods till input is exhausted

  % read one period correcting for Doppler
  %
  [demods, Eflag] = hdedop(samples, VonC, FConFS, start_scan, n_scans);

  if Eflag == 0                       % process if a set of samples was read
    PeriodCount = PeriodCount + 1;   % increment count of periods
    start_scan = start_scan+n_scans; % set up start of next period read

    % apply sinc filter wrapping around full period
    %
    temp = zeros(size(demods));
    N = length(demods);
    for count = 0:Qfilter-1
      temp = temp+demods(mod([0:N-1]+count, N)+1);
    end
    demods = temp/Qfilter; temp=[];  % and normalize

    % do sequence removal keeping only resulting amplitudes
    %
    tseries(:,PeriodCount) = abs(fmst(demods, perm, angle));

  end
end

% write out the processed data set and the name of the input file
%
save plot tseries fname;

fprintf('Number of sequence periods processed: %d\n', PeriodCount);
toc          % print total execution time
```

4.4.2 hiftwfp.m

This script was used to generate to plot shown in Figure 2.

```
% hiftwfp.m
%
% script to generate waterfall plot of pre-processed
% data set.  Assumes one-half periods of skipped in
% the processing to match the processing done in early
% 1992.
%
% 6June2000...initial version...K.Metzger
%
clear all;

% specify starting bin and number of bins to plot
%
start_bin = 5600; % matches x0270949.sam processing
plot_bins = 2001;

load plot.mat tseries fname;     % read pre-processed data set

[nrows, ncols] = size(tseries);  % determine data set size

% extract only portion of most interest
%
tseries = tseries(start_bin:start_bin+plot_bins-1, :);

[nrows, ncols] = size(tseries);  % determine subset size

figure =1; clf; hold on;

% generate the grid MATLAB needs for waterfall plots
%
[X, Y] = meshgrid(start_bin:start_bin+nrows-1, 1:ncols);

tseries = tseries.*tseries;  % magnitudes squared

zmax = max(max(tseries));    % to scale the plotted values

% generate waterfall and setup desired plot characteristics
%
colormap([0 0 0]);
waterfall(X, Y, tseries.');
axis([start_bin, start_bin+nrows-1, 1, ncols, 1, 2.5*zmax]);
view(0, 60);
grid off;
title(fname);
```

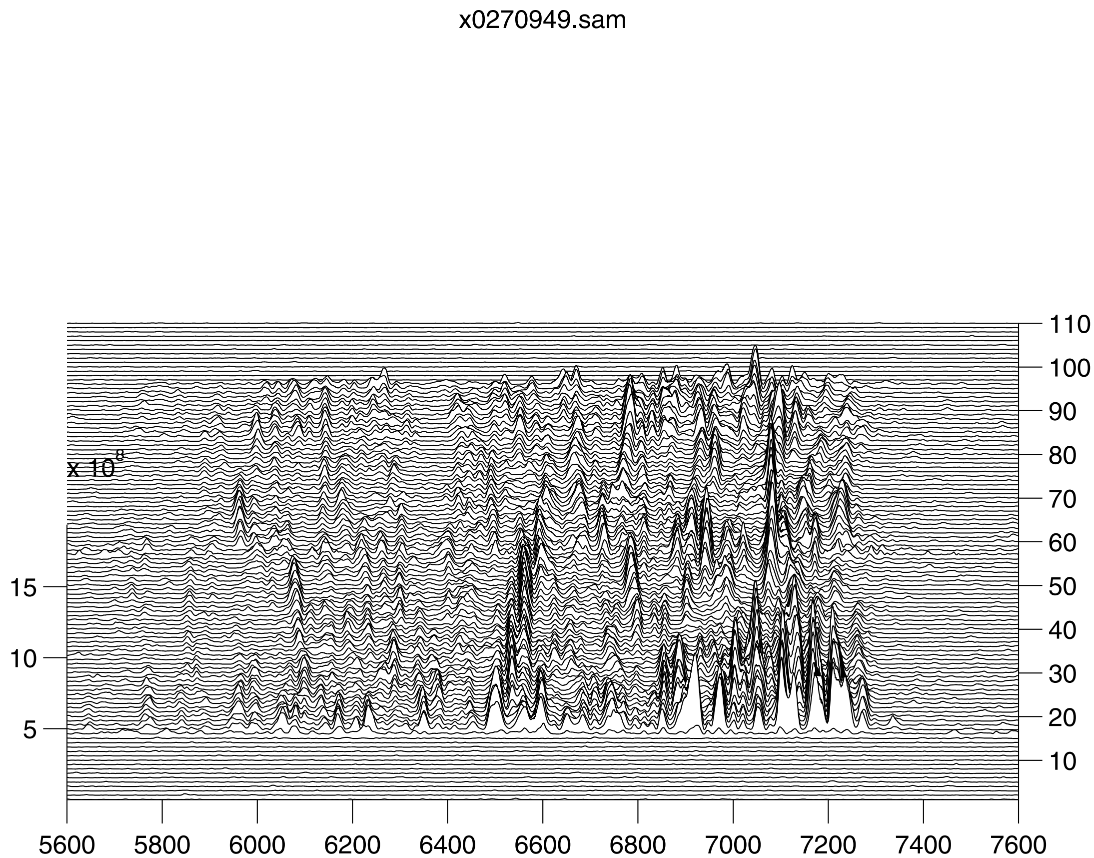

Figure 2: Test plot of the main portion of the multipath structure obtained using data set x0270949.sam.

5 Transmissions

5.1 Transmitted waveforms

The following record was generated by Matthew Dzieciuch then of the University of Michigan and presently at Scripps. The original file is contained on CD 1 and is named `hift.readme`.

The source was a 10 element array centered on the sound channel axis. The sound channel axis was at a depth of 175 meters. The array spacing was at 12.5 feet. Only 5 sources or less were on at any particular time. Numbering starts from the top of the array.

Date	Time	Signal	Sources	Ship Speed (m/s)	Heading (degrees true)
26 january	0000:5	CW	1 3 5	1.50	9.58
26 january	0002	CW	1 3 5 7		

Date	Time	Signal	Sources	Ship Speed (m/s)	Heading (degrees true)
26 january	0050:5	CW			
26 january	0053	CW	1 3 5 7		
26 january	0100	CW			
26 january	0300	P	1 3 5 7 10	1.61	320.70
26 january	0400	P			
26 january	0600	M1	2 4 6 8 10	1.64	300.08
26 january	0700	M1			
26 january	0900	M3	2 4 6 8 10	1.58	298.93
26 january	1000	M3			
26 january	1200	CW	2 4 6 8 10	1.54	254.53
26 january	1300	CW			
26 january	1500	P	2 4 6 8 10	1.54	232.71
26 january	1525	P	4 6 8 10		
26 january	1600	P			
26 january	1800	M1	1 4 6 8 10	1.60	224.96
26 january	1819	M1	3 4 6 8 10		
26 january	1900	M1			
26 january	2100	M4	3 4 6 8 10	1.34	251.67
26 january	2200	M4			
27 january	0000	CW	3 4 6 8 10	1.29	254.09
27 january	0100	CW			
27 january	0300	P	3 4 6 8 10	1.92	243.69
27 january	0400	P			
27 january	0600	M1	3 4 6 8	1.50	239.70
27 january	0601	M1	3 4 6 8 10		
27 january	0700	M1			
27 january	0900	M2	3 4 6 10	1.81	238.66
27 january	0902	M2	1 3 4 6 10		
27 january	0918	M2	3 4 6 10		
27 january	0950	M2	3 4 6 7 10		
27 january	1000	M2			
27 january	1200	CW	1 3 4 6 10	1.71	234.36
27 january	1204	CW	1 3 4 6		
27 january	1205	CW	1 3 4 6 7		
27 january	1215	CW	3 4 6 7		
27 january	1230	CW	3 4 5 6 7		
27 january	1300	CW			
27 january	1500	P	unknown	1.55	237.90
27 january	1600	P			

Date	Time	Signal	Sources	Ship Speed (m/s)	Heading (degrees true)
29 january	0900	M3	2 3 4 5 9	1.68	178.91
29 january	0910	M3	3 4 5 9		
29 january	0912	M3	3 4 5 6 9		
29 january	0914	M3	intermittent	on all sources	
29 january	0916	M3	3 4 5 6 9		
29 january	1000	M3			
29 january	1200	CW	3 4 5 6 9	1.77	172.96
29 january	1300	CW			
29 january	1500	P	3 4 5 6 9	1.81	180.95
29 january	1600	P			
29 january	1800	M1	3 4 5 6 9	1.74	183.96
29 january	1837	M1	3 4 5 6		
29 january	1900	M1			
29 january	2100	M4	3 4 5 6	1.98	328.91
29 january	2200	M4			
30 january	0000	CW	3 4 5 6	1.03	335.
30 january	0100	CW			
30 january	0300	P	3 4 5 6	1.49	313.
30 january	0400	P			
30 january	0600	M1	3 4 5 6	1.59	300.
30 january	0700	M1			
30 january	0900	M2	3 5	1.70	270.
30 january	0940	M2	3 6		
30 january	1000	M2			
30 january	1200	CW	3 6	2.09	270.63
30 january	1300	CW			
30 january	1500	P	4 6	1.61	268.28
30 january	1600	P			
30 january	1800	M1	4 6	1.77	273.07
30 january	1900	M1			
30 january	2100	M3	3 4 6	1.44	280.29
30 january	2117	M3	3 6		
30 january	2200	M3			
31 january	0000	CW	6	1.16	289.17
31 january	0030	CW			
31 january	0300	P	6	1.44	10.
31 january	0330	P			

Date	Time	Signal	Sources	Ship Speed (m/s)	Heading (degrees true)
31 january	0600	M1	6	2.06	327.
31 january	0630	M1			
31 january	0900	M4	6	1.59	328.
31 january	0930	M4			
31 january	1213	CW	3	1.27	266.41
31 january	1230	CW			
31 january	1500	P	3	1.91	267.84
31 january	1530	P			
31 january	1800	M1	3	1.41	273.48
31 january	1830	M1			
31 january	2100	M2	3	1.62	277.28
31 january	2130	M2			

The local sound speed profile measured by an XSV was:

depth (meters)	speed (meters/second)
0	1459.7
25	1461.1
50	1461.3
75	1461.1
100	1458.4
125	1453.8
150	1453.6
175	1453.3
200	1454.3
225	1454.5
250	1455.5
300	1460.0
350	1462.0
400	1463.3
450	1465.0
500	1466.2
550	1467.2
600	1468.2
700	1469.9
800	1471.6
1000	1474.5

The water depth at the source was 1150 meters.

5.2 GPS tracking of the source ship location

A GPS receiver was available on the source ship and was used to log the ship position. The resulting files are contained on CD 1.

6 Data sets and associated Doppler estimates

The Doppler estimates given below used sound speed values of 2913 knots and 1498 m/s to convert estimated carrier frequency offsets to ship velocity. These values are based on estimates made of the average frequency shift observed over each reception. They are useful when working with the data sets but it should be kept in mind that other estimation procedures likely will result in other values.

For reference, the nautical mile is defined as 1852 meters.

6.1 Data file format

Data files consist of an ASCII header followed by binary data generated by A/D scans. The header starts with the line *HEADER* and is terminated by the line *DATA*. A single LF (OA hex) character terminates each line in the header. Sample values follow immediately, are 16 bits in size and use the INTEL byte ordering

An example of a header from a single channel data set is: .

```
*HEADER*
Takedata: Heard Island A/D Spooler 23November90
sample file: x0270949.sam
site:x  n_chans: 1  freq: 57 Hz
source   lat/lon: 53 deg  13.9 min S   74 deg  30.6 min E
receiver lat/lon: 10 deg  30.0 min S  105 deg  35.0 min E
distance:  2967.6 nm, propagation time: 61 min  7 sec
timer tics:200
  channel: 0   gain: 1
filter board gain:61
started waiting Sun Jan 27 09:49:56 1991
Julian:027:09:49:56
*DATA*
```

An example of a header from a multiple channel data set is:

```
*HEADER*
Takedata: Heard Island A/D Spooler 18January91
sample file: data\b0302352.sam
site:b  n_chans: 8  freq: 57 Hz
source   lat/lon: 53 deg  13.9 min S   74 deg  30.6 min E
```

```
receiver lat/lon:  0 deg   0.0 min N    0 deg   0.0 min E
distance:  8933.2 nm, propagation time:184 min  0 sec
timer tics:200
  channel: 0   gain: 1
  channel: 1   gain: 1
  channel: 2   gain: 1
  channel: 3   gain: 1
  channel: 4   gain: 1
  channel: 5   gain: 1
  channel: 6   gain: 1
  channel: 7   gain: 1
filter board gain: 2
started waiting Wed Jan 30 23:52:16 1991
Julian:030:23:52:16
*DATA*
```

The Russian seance files do not include a header. These files were reformatted with a minimal header added. An example is:

```
*HEADER*
sample file: seance08.ch3
site: r  n_chans: 1  freq: 57 Hz  clock_factor: 4
 timer tics:20
  channel: 0   gain: 1
started waiting Sat Dec  6 09:46:48 1997
*DATA*
```

6.2 Ascension Island

Data set 0 is contained on CD #2 in the directory **A1**..

Doppler estimates for Ascension Island data set 0.

file	channel	m/s	knots	signal
a0260130.sam	1	-0.313	-0.608	CW
a0260130.sam	2	-0.342	-0.666	CW
a0260130.sam	3	-0.338	-0.657	CW
a0260130.sam	4	-0.319	-0.620	CW
a0260430.sam	1	1.021	1.985	P
a0260430.sam	2	1.035	2.012	P
a0260430.sam	3	1.029	2.000	P
a0260430.sam	4	1.019	1.980	P
a0260730.sam	1	1.448	2.815	M1

Doppler estimates for Ascension Island data set 0.

file	channel	m/s	knots	signal
a0260730.sam	2	1.453	2.824	M1
a0260730.sam	3	1.447	2.813	M1
a0260730.sam	4	1.462	2.841	M1
a0261030.sam	1	1.431	2.782	M3
a0261030.sam	2	1.419	2.759	M3
a0261030.sam	3	1.421	2.762	M3
a0261030.sam	4	1.424	2.769	M3
a0261330.sam	1	1.534	2.983	CW
a0261330.sam	2	1.536	2.986	CW
a0261330.sam	3	1.542	2.997	CW
a0261330.sam	4	1.529	2.971	CW
a0261657.sam	1	1.339	2.603	P
a0261657.sam	2	1.318	2.562	P
a0261657.sam	3	1.324	2.574	P
a0261657.sam	4	1.327	2.579	P
a0261930.sam	1	1.190	2.314	M1
a0261930.sam	2	1.185	2.304	M1
a0261930.sam	3	1.188	2.310	M1
a0261930.sam	4	1.192	2.318	M1
a0262230.sam	1	1.339	2.602	M4
a0262230.sam	2	1.340	2.605	M4
a0262230.sam	3	1.339	2.603	M4
a0262230.sam	4	1.340	2.605	M4
a0270130.sam	1	1.257	2.444	CW
a0270130.sam	2	1.296	2.520	CW
a0270130.sam	3	1.296	2.519	CW
a0270130.sam	4	1.267	2.463	CW
a0270430.sam	1	1.801	3.501	P
a0270430.sam	2	1.799	3.496	P
a0270430.sam	3	1.792	3.484	P
a0270430.sam	4	1.787	3.474	P
a0270730.sam	1	1.363	2.649	M1
a0270730.sam	2	1.383	2.689	M1
a0270730.sam	3	1.369	2.662	M1
a0270730.sam	4	1.371	2.665	M1
a0271030.sam	1	1.626	3.161	M2
a0271030.sam	2	1.632	3.173	M2

Doppler estimates for Ascension Island data set 0.

file	channel	m/s	knots	signal
a0271030.sam	3	1.632	3.172	M2
a0271030.sam	4	1.630	3.169	M2
a0271330.sam	1	1.466	2.849	CW
a0271330.sam	2	1.487	2.891	CW
a0271330.sam	3	1.480	2.877	CW
a0271330.sam	4	1.477	2.872	CW
a0271630.sam	1	1.391	2.703	P
a0271630.sam	2	1.404	2.728	P
a0271630.sam	3	1.397	2.715	P
a0271630.sam	4	1.398	2.718	P
a0291030.sam	1	0.055	0.107	M3
a0291030.sam	2	0.074	0.144	M3
a0291030.sam	3	0.070	0.137	M3
a0291030.sam	4	0.050	0.097	M3
a0291330.sam	1	-0.135	-0.262	CW
a0291330.sam	2	-0.137	-0.266	CW
a0291330.sam	3	-0.133	-0.259	CW
a0291330.sam	4	-0.154	-0.299	CW
a0291630.sam	1	0.074	0.143	P
a0291630.sam	2	0.082	0.159	P
a0291630.sam	3	0.082	0.160	P
a0291630.sam	4	0.089	0.172	P
a0291930.sam	1	0.154	0.299	M1
a0291930.sam	2	0.145	0.282	M1
a0291930.sam	3	0.173	0.336	M1
a0291930.sam	4	0.165	0.321	M1
a0292230.sam	1	1.016	1.974	M4
a0292230.sam	2	1.009	1.961	M4
a0292230.sam	3	1.015	1.973	M4
a0292230.sam	4	1.031	2.003	M4
a0300130.sam	1	0.436	0.847	CW
a0300130.sam	2	0.422	0.821	CW
a0300130.sam	3	0.426	0.828	CW
a0300130.sam	4	0.434	0.843	CW
a0300430.sam	1	1.102	2.141	P
a0300430.sam	2	1.103	2.144	P
a0300430.sam	3	1.110	2.157	P

Doppler estimates for Ascension Island data set 0.

file	channel	m/s	knots	signal
a0300430.sam	4	1.102	2.143	P
a0300730.sam	1	1.485	2.886	M1
a0300730.sam	2	1.529	2.972	M1
a0300730.sam	3	1.463	2.844	M1
a0300730.sam	4	1.483	2.882	M1
a0300755.sam	1	1.404	2.729	M1
a0300755.sam	2	1.404	2.730	M1
a0300755.sam	3	1.402	2.725	M1
a0300755.sam	4	1.398	2.717	M1
a0301030.sam	1	1.750	3.402	M2
a0301030.sam	2	1.759	3.418	M2
a0301030.sam	3	1.771	3.442	M2
a0301030.sam	4	1.764	3.428	M2
a0301330.sam	1	2.156	4.191	CW
a0301330.sam	2	2.140	4.160	CW
a0301330.sam	3	2.147	4.174	CW
a0301330.sam	4	2.143	4.166	CW
a0301630.sam	1	1.663	3.232	P
a0301630.sam	2	1.636	3.181	P
a0301630.sam	3	1.659	3.224	P
a0301630.sam	4	1.661	3.228	P
a0301930.sam	1	1.805	3.509	M1
a0301930.sam	2	1.815	3.528	M1
a0301930.sam	3	1.816	3.531	M1
a0301930.sam	4	1.812	3.523	M1
a0302230.sam	1	1.483	2.882	M3
a0302230.sam	2	1.479	2.874	M3
a0302230.sam	3	1.458	2.833	M3
a0302230.sam	4	1.482	2.881	M3
a0310130.sam	1	1.102	2.141	CW
a0310130.sam	2	1.103	2.144	CW
a0310130.sam	3	1.100	2.138	CW
a0310130.sam	4	1.110	2.158	CW
a0310430.sam	1	-2.927	-5.689	P
a0310430.sam	2	2.927	5.690	P
a0310430.sam	3	-2.927	-5.690	P

Doppler estimates for Ascension Island data set 0.

file	channel	m/s	knots	signal
a0310430.sam	4	-2.927	-5.689	P
a0310730.sam	1	1.095	2.128	M1
a0310730.sam	2	1.058	2.056	M1
a0310730.sam	3	1.050	2.041	M1
a0310730.sam	4	1.054	2.049	M1
a0311030.sam	1	0.811	1.576	M4
a0311030.sam	2	0.801	1.556	M4
a0311030.sam	3	0.784	1.524	M4
a0311030.sam	4	0.783	1.522	M4
a0311330.sam	1	1.335	2.595	CW
a0311330.sam	2	1.364	2.652	CW
a0311330.sam	3	1.374	2.671	CW
a0311330.sam	4	1.364	2.652	CW
a0311630.sam	1	1.927	3.746	P
a0311630.sam	2	1.958	3.805	P
a0311630.sam	3	2.087	4.058	P
a0311630.sam	4	1.901	3.695	P
a0311930.sam	1	-1.429	-2.777	M1
a0311930.sam	2	1.437	2.793	M1
a0311930.sam	3	1.537	2.988	M1
a0311930.sam	4	1.536	2.985	M1
a0312230.sam	1	1.574	3.060	M2
a0312230.sam	2	1.698	3.300	M2
a0312230.sam	3	1.539	2.992	M2
a0312230.sam	4	1.719	3.341	M2
a0320130.sam	1	1.824	3.546	NONE
a0320130.sam	2	2.208	4.291	NONE
a0320130.sam	3	1.828	3.553	NONE
a0320130.sam	4	1.508	2.932	NONE

Doppler estimates for Ascension Island data set 1.

file	channel	m/s	knots	signal
a1260130.sam	1	-0.336	-0.654	CW
a1260130.sam	2	-0.332	-0.646	CW
a1260130.sam	3	-0.338	-0.658	CW

Doppler estimates for Ascension Island data set 1.

file	channel	m/s	knots	signal
a1260130.sam	4	-0.333	-0.647	CW
a1260430.sam	1	1.014	1.970	P
a1260430.sam	2	1.034	2.010	P
a1260430.sam	3	1.024	1.991	P
a1260430.sam	4	1.020	1.983	P
a1260730.sam	1	1.444	2.807	M1
a1260730.sam	2	1.462	2.841	M1
a1260730.sam	3	1.473	2.863	M1
a1260730.sam	4	1.438	2.795	M1
a1261030.sam	1	1.398	2.718	M3
a1261030.sam	2	1.437	2.794	M3
a1261030.sam	3	1.431	2.782	M3
a1261030.sam	4	1.409	2.739	M3
a1261330.sam	1	1.542	2.997	CW
a1261330.sam	2	1.548	3.009	CW
a1261330.sam	3	1.530	2.975	CW
a1261330.sam	4	1.502	2.920	CW
a1261638.sam	1	1.339	2.602	P
a1261638.sam	2	1.197	2.328	P
a1261638.sam	3	1.310	2.546	P
a1261638.sam	4	1.383	2.688	P
a1261930.sam	1	1.199	2.330	M1
a1261930.sam	2	1.157	2.248	M1
a1261930.sam	3	1.195	2.324	M1
a1261930.sam	4	1.192	2.318	M1
a1262230.sam	1	1.358	2.640	M4
a1262230.sam	2	1.390	2.703	M4
a1262230.sam	3	1.346	2.617	M4
a1262230.sam	4	1.381	2.684	M4
a1270130.sam	1	1.307	2.540	CW
a1270130.sam	2	1.293	2.514	CW
a1270130.sam	3	1.301	2.528	CW
a1270130.sam	4	1.221	2.373	CW
a1270430.sam	1	1.813	3.524	P
a1270430.sam	2	1.786	3.472	P
a1270430.sam	3	1.793	3.485	P

Doppler estimates for Ascension Island data set 1.

file	channel	m/s	knots	signal
a1270430.sam	4	1.812	3.522	P
a1270730.sam	1	1.380	2.682	M1
a1270730.sam	2	1.376	2.675	M1
a1270730.sam	3	1.377	2.677	M1
a1270730.sam	4	1.402	2.726	M1
a1271330.sam	1	1.494	2.904	CW
a1271330.sam	2	1.469	2.856	CW
a1271330.sam	3	1.487	2.891	CW
a1271330.sam	4	1.496	2.909	CW
a1271630.sam	1	1.415	2.751	P
a1271630.sam	2	1.402	2.725	P
a1271630.sam	3	1.410	2.741	P
a1271630.sam	4	1.412	2.745	P
a1291030.sam	1	0.084	0.163	M3
a1291030.sam	2	0.047	0.091	M3
a1291030.sam	3	0.060	0.116	M3
a1291030.sam	4	0.064	0.124	M3
a1291330.sam	1	-0.114	-0.222	CW
a1291330.sam	2	-0.127	-0.246	CW
a1291330.sam	3	-0.143	-0.278	CW
a1291330.sam	4	-0.110	-0.214	CW
a1291630.sam	1	0.118	0.229	P
a1291630.sam	2	0.081	0.158	P
a1291630.sam	3	0.070	0.135	P
a1291630.sam	4	0.077	0.150	P
a1291930.sam	1	0.202	0.393	M1
a1291930.sam	2	0.123	0.240	M1
a1291930.sam	3	0.149	0.290	M1
a1291930.sam	4	0.139	0.271	M1
a1292230.sam	1	0.984	1.913	M4
a1292230.sam	2	1.023	1.988	M4
a1292230.sam	3	1.022	1.986	M4
a1292230.sam	4	1.012	1.966	M4
a1300130.sam	1	0.396	0.771	CW
a1300130.sam	2	0.424	0.825	CW
a1300130.sam	3	0.421	0.819	CW

Doppler estimates for Ascension Island data set 1.

file	channel	m/s	knots	signal
a1300130.sam	4	0.409	0.796	CW
a1300430.sam	1	1.068	2.076	P
a1300430.sam	2	1.095	2.129	P
a1300430.sam	3	1.109	2.157	P
a1300430.sam	4	1.092	2.123	P
a1300730.sam	1	1.406	2.732	M1
a1300730.sam	2	1.427	2.773	M1
a1300730.sam	3	1.408	2.737	M1
a1300730.sam	4	1.410	2.741	M1
a1301030.sam	1	1.775	3.450	M2
a1301030.sam	2	1.815	3.529	M2
a1301030.sam	3	1.762	3.425	M2
a1301030.sam	4	1.793	3.486	M2
a1301330.sam	1	2.156	4.192	CW
a1301330.sam	2	2.162	4.202	CW
a1301330.sam	3	2.149	4.178	CW
a1301330.sam	4	2.177	4.232	CW
a1301630.sam	1	1.662	3.231	P
a1301630.sam	2	1.522	2.959	P
a1301630.sam	3	1.658	3.222	P
a1301630.sam	4	1.557	3.027	P
a1301930.sam	1	1.807	3.512	M1
a1301930.sam	2	1.779	3.459	M1
a1301930.sam	3	1.813	3.524	M1
a1301930.sam	4	1.788	3.476	M1
a1302230.sam	1	1.474	2.864	M3
a1302230.sam	2	1.458	2.834	M3
a1302230.sam	3	1.478	2.873	M3
a1302230.sam	4	1.466	2.850	M3
a1310130.sam	1	1.122	2.181	CW
a1310130.sam	2	1.074	2.087	CW
a1310130.sam	3	1.114	2.166	CW
a1310130.sam	4	1.106	2.151	CW
a1310430.sam	1	-0.118	-0.229	P
a1310430.sam	2	2.166	4.210	P
a1310430.sam	3	-0.109	-0.211	P

Doppler estimates for Ascension Island data set 1.

file	channel	m/s	knots	signal
a1310430.sam	4	-2.231	-4.336	P
a1310730.sam	1	1.056	2.052	M1
a1310730.sam	2	1.081	2.102	M1
a1310730.sam	3	1.078	2.095	M1
a1310730.sam	4	0.785	1.526	M1
a1311030.sam	1	0.771	1.499	M4
a1311030.sam	2	1.818	3.534	M4
a1311030.sam	3	0.782	1.519	M4
a1311030.sam	4	0.748	1.455	M4
a1311330.sam	1	1.378	2.679	CW
a1311330.sam	2	1.439	2.798	CW
a1311330.sam	3	1.377	2.677	CW
a1311330.sam	4	1.438	2.796	CW
a1311630.sam	1	1.939	3.770	P
a1311630.sam	2	-1.113	-2.164	P
a1311630.sam	3	2.051	3.988	P
a1311630.sam	4	1.131	2.199	P
a1311930.sam	1	1.455	2.829	M1
a1311930.sam	2	-1.456	-2.831	M1
a1311930.sam	3	1.430	2.779	M1
a1311930.sam	4	1.483	2.883	M1
a1312230.sam	1	1.749	3.399	M2
a1312230.sam	2	2.566	4.988	M2
a1312230.sam	3	1.756	3.413	M2
a1312230.sam	4	2.235	4.344	M2
a1320130.sam	1	0.738	1.435	NONE
a1320130.sam	2	1.819	3.537	NONE
a1320130.sam	3	3.292	6.399	NONE
a1320130.sam	4	-2.190	-4.256	NONE

6.3 Bermuda

Doppler estimates for Bermuda data set part 1.

file	channel	m/s	knots	signal
b0260248.sam	1	-0.634	-1.233	CW

Doppler estimates for Bermuda data set part 1.

file	channel	m/s	knots	signal
b0260248.sam	2	-0.624	-1.214	CW
b0260248.sam	3	-0.640	-1.245	CW
b0260248.sam	4	-0.686	-1.334	CW
b0260248.sam	5	-0.350	-0.681	CW
b0260248.sam	6	-0.360	-0.700	CW
b0260248.sam	7	-0.654	-1.271	CW
b0260248.sam	8	-0.646	-1.255	CW
b0260548.sam	1	0.743	1.444	P
b0260548.sam	2	0.752	1.462	P
b0260548.sam	3	0.715	1.390	P
b0260548.sam	4	0.703	1.366	P
b0260548.sam	5	0.965	1.875	P
b0260548.sam	6	1.017	1.977	P
b0260548.sam	7	0.726	1.411	P
b0260548.sam	8	0.736	1.430	P
b0260848.sam	1	1.222	2.375	M1
b0260848.sam	2	1.228	2.386	M1
b0260848.sam	3	1.415	2.750	M1
b0260848.sam	4	1.213	2.358	M1
b0260848.sam	5	1.223	2.378	M1
b0260848.sam	6	1.424	2.768	M1
b0260848.sam	7	1.457	2.832	M1
b0261148.sam	1	1.202	2.336	M3
b0261148.sam	2	1.188	2.309	M3
b0261148.sam	3	1.379	2.680	M3
b0261148.sam	4	1.202	2.336	M3
b0261148.sam	5	1.198	2.328	M3
b0261148.sam	6	1.476	2.869	M3
b0261148.sam	7	1.389	2.701	M3
b0261448.sam	1	1.588	3.088	CW
b0261448.sam	2	1.585	3.080	CW
b0261448.sam	3	1.607	3.123	CW
b0261448.sam	4	1.588	3.087	CW
b0261448.sam	5	1.557	3.026	CW
b0261448.sam	6	1.731	3.365	CW
b0261448.sam	7	1.503	2.922	CW
b0261748.sam	1	1.454	2.827	P
b0261748.sam	2	1.468	2.853	P

Doppler estimates for Bermuda data set part 1.

file	channel	m/s	knots	signal
b0261748.sam	3	1.371	2.664	P
b0261748.sam	4	1.435	2.790	P
b0261748.sam	5	1.460	2.838	P
b0261748.sam	6	1.413	2.746	P
b0261748.sam	7	1.354	2.632	P
b0262048.sam	1	1.390	2.703	M1
b0262048.sam	2	1.384	2.689	M1
b0262048.sam	3	1.205	2.342	M1
b0262048.sam	4	1.391	2.703	M1
b0262048.sam	5	1.386	2.693	M1
b0262048.sam	6	1.438	2.796	M1
b0262048.sam	7	1.177	2.289	M1
b0262348.sam	1	1.407	2.735	M4
b0262348.sam	2	1.411	2.743	M4
b0262348.sam	3	1.387	2.696	M4
b0262348.sam	4	1.412	2.744	M4
b0262348.sam	5	1.408	2.736	M4
b0262348.sam	6	-0.019	-0.037	M4
b0262348.sam	7	1.346	2.616	M4
b0270248.sam	1	1.368	2.660	CW
b0270248.sam	2	1.359	2.643	CW
b0270248.sam	3	1.357	2.639	CW
b0270248.sam	4	1.361	2.645	CW
b0270248.sam	5	1.360	2.643	CW
b0270248.sam	6	1.439	2.797	CW
b0270248.sam	7	1.325	2.576	CW
b0270548.sam	1	1.915	3.721	P
b0270548.sam	2	1.930	3.751	P
b0270548.sam	3	1.844	3.584	P
b0270548.sam	4	1.935	3.762	P
b0270548.sam	5	1.929	3.750	P
b0270548.sam	6	1.767	3.434	P
b0270548.sam	7	1.815	3.529	P
b0270848.sam	1	1.483	2.883	M1
b0270848.sam	2	1.487	2.891	M1
b0270848.sam	3	1.399	2.719	M1
b0270848.sam	4	1.473	2.863	M1
b0270848.sam	5	1.471	2.859	M1

Doppler estimates for Bermuda data set part 1.

file	channel	m/s	knots	signal
b0270848.sam	6	1.458	2.834	M1
b0270848.sam	7	1.387	2.696	M1
b0271148.sam	1	1.792	3.483	M2
b0271148.sam	2	1.808	3.515	M2
b0271148.sam	3	1.689	3.283	M2
b0271148.sam	4	1.792	3.483	M2
b0271148.sam	5	1.806	3.511	M2
b0271148.sam	6	1.765	3.431	M2
b0271148.sam	7	1.618	3.144	M2
b0271448.sam	1	1.607	3.125	CW
b0271448.sam	2	1.670	3.247	CW
b0271448.sam	3	1.542	2.998	CW
b0271448.sam	4	1.623	3.155	CW
b0271448.sam	5	1.659	3.224	CW
b0271448.sam	6	1.542	2.997	CW
b0271448.sam	7	1.489	2.894	CW
b0271748.sam	1	1.517	2.948	P
b0271748.sam	2	1.536	2.985	P
b0271748.sam	3	1.426	2.772	P
b0271748.sam	4	1.530	2.974	P
b0271748.sam	5	1.510	2.934	P
b0271748.sam	6	1.398	2.717	P
b0271748.sam	7	1.416	2.753	P
b0272048.sam	1	-2.570	-4.996	NONE
b0272048.sam	2	-3.230	-6.278	NONE
b0272048.sam	3	1.536	2.985	NONE
b0272048.sam	4	-2.119	-4.118	NONE
b0272048.sam	5	3.256	6.329	NONE
b0272048.sam	6	-0.732	-1.422	NONE
b0272048.sam	7	-3.303	-6.420	NONE
b0272348.sam	1	-2.195	-4.267	NONE
b0272348.sam	2	-3.289	-6.393	NONE
b0272348.sam	3	1.835	3.568	NONE
b0272348.sam	4	-1.838	-3.573	NONE
b0272348.sam	5	3.331	6.475	NONE
b0272348.sam	6	-3.258	-6.333	NONE
b0272348.sam	7	-3.339	-6.491	NONE
b0290548.sam	1	-3.280	-6.375	NONE

Doppler estimates for Bermuda data set part 1.

file	channel	m/s	knots	signal
b0290548.sam	2	-2.524	-4.907	NONE
b0290548.sam	3	1.606	3.121	NONE
b0290548.sam	4	3.311	6.436	NONE
b0290548.sam	5	-3.273	-6.361	NONE
b0290548.sam	6	-3.292	-6.399	NONE
b0290548.sam	7	-3.293	-6.401	NONE
b0290548.sam	8	1.120	2.176	NONE
b0290848.sam	1	-3.310	-6.434	NONE
b0290848.sam	2	-1.816	-3.530	NONE
b0290848.sam	3	2.559	4.975	NONE
b0290848.sam	4	2.927	5.689	NONE
b0290848.sam	5	-3.310	-6.434	NONE
b0290848.sam	6	2.892	5.622	NONE
b0290848.sam	7	-3.314	-6.441	NONE
b0290848.sam	8	2.561	4.978	NONE
b0291148.sam	1	0.394	0.767	M3
b0291148.sam	2	0.414	0.804	M3
b0291148.sam	3	0.112	0.218	M3
b0291148.sam	4	0.397	0.771	M3
b0291148.sam	5	0.412	0.802	M3
b0291148.sam	6	0.298	0.579	M3
b0291148.sam	7	0.069	0.133	M3
b0291148.sam	8	0.104	0.202	M3
b0291452.sam	1	0.240	0.466	CW
b0291452.sam	2	0.268	0.521	CW
b0291452.sam	3	-0.049	-0.095	CW
b0291452.sam	4	0.239	0.465	CW
b0291452.sam	5	0.276	0.537	CW
b0291452.sam	6	-0.010	-0.020	CW
b0291452.sam	7	-0.076	-0.148	CW
b0291452.sam	8	-0.026	-0.050	CW
b0291752.sam	1	0.451	0.877	P
b0291752.sam	2	0.485	0.942	P
b0291752.sam	3	0.245	0.477	P
b0291752.sam	4	0.468	0.909	P
b0291752.sam	5	0.470	0.913	P
b0291752.sam	6	0.301	0.586	P
b0291752.sam	7	0.104	0.202	P

Doppler estimates for Bermuda data set part 1.

file	channel	m/s	knots	signal
b0291752.sam	8	0.305	0.592	P
b0292052.sam	1	0.490	0.952	M1
b0292052.sam	2	1.850	3.597	M1
b0292052.sam	3	0.293	0.569	M1
b0292052.sam	4	0.511	0.994	M1
b0292052.sam	5	-2.592	-5.038	M1
b0292052.sam	6	0.368	0.715	M1
b0292052.sam	7	0.234	0.454	M1
b0292052.sam	8	0.319	0.620	M1
b0292352.sam	1	0.634	1.232	M4
b0292352.sam	2	0.608	1.181	M4
b0292352.sam	3	0.871	1.693	M4
b0292352.sam	4	0.650	1.263	M4
b0292352.sam	5	0.635	1.234	M4
b0292352.sam	6	0.763	1.484	M4
b0292352.sam	7	1.015	1.973	M4
b0292352.sam	8	0.801	1.557	M4

Doppler estimates for Bermuda data set part 2.

file	channel	m/s	knots	signal
b0300252.sam	1	0.268	0.520	CW
b0300252.sam	2	0.230	0.447	CW
b0300252.sam	3	0.377	0.733	CW
b0300252.sam	4	0.253	0.492	CW
b0300252.sam	5	0.240	0.466	CW
b0300252.sam	6	0.358	0.695	CW
b0300252.sam	7	0.448	0.871	CW
b0300252.sam	8	0.369	0.718	CW
b0300552.sam	1	0.841	1.635	P
b0300552.sam	2	0.833	1.620	P
b0300552.sam	3	1.080	2.099	P
b0300552.sam	4	0.826	1.606	P
b0300552.sam	5	0.770	1.497	P
b0300552.sam	6	1.098	2.134	P
b0300552.sam	7	1.116	2.170	P
b0300552.sam	8	1.050	2.041	P
b0300852.sam	1	1.207	2.346	M1

Doppler estimates for Bermuda data set part 2.

file	channel	m/s	knots	signal
b0300852.sam	2	1.164	2.263	M1
b0300852.sam	3	1.420	2.759	M1
b0300852.sam	4	1.176	2.285	M1
b0300852.sam	5	1.122	2.181	M1
b0300852.sam	6	1.489	2.894	M1
b0300852.sam	7	1.420	2.760	M1
b0300852.sam	8	1.384	2.690	M1
b0301152.sam	1	1.720	3.344	M2
b0301152.sam	2	1.708	3.320	M2
b0301152.sam	3	1.764	3.429	M2
b0301152.sam	4	1.722	3.348	M2
b0301152.sam	5	1.771	3.443	M2
b0301152.sam	6	-0.065	-0.127	M2
b0301152.sam	7	1.791	3.482	M2
b0301152.sam	8	1.783	3.466	M2
b0301452.sam	1	2.095	4.073	CW
b0301452.sam	2	2.104	4.091	CW
b0301452.sam	3	2.164	4.206	CW
b0301452.sam	4	2.105	4.091	CW
b0301452.sam	5	2.150	4.178	CW
b0301452.sam	6	-3.250	-6.318	CW
b0301452.sam	7	2.173	4.225	CW
b0301452.sam	8	2.160	4.199	CW
b0301752.sam	1	1.658	3.223	P
b0301752.sam	2	1.612	3.134	P
b0301752.sam	3	1.697	3.299	P
b0301752.sam	4	1.601	3.112	P
b0301752.sam	5	1.742	3.387	P
b0301752.sam	6	-3.293	-6.401	P
b0301752.sam	7	1.696	3.297	P
b0301752.sam	8	1.732	3.366	P
b0302052.sam	1	1.751	3.404	M1
b0302052.sam	2	1.730	3.363	M1
b0302052.sam	3	1.803	3.504	M1
b0302052.sam	4	1.775	3.449	M1
b0302052.sam	5	1.753	3.408	M1
b0302052.sam	6	-0.766	-1.490	M1
b0302052.sam	7	1.832	3.560	M1

Doppler estimates for Bermuda data set part 2.

file	channel	m/s	knots	signal
b0302052.sam	8	1.783	3.467	M1
b0302352.sam	1	1.386	2.695	M3
b0302352.sam	2	1.380	2.682	M3
b0302352.sam	3	1.447	2.813	M3
b0302352.sam	4	1.402	2.724	M3
b0302352.sam	5	1.425	2.771	M3
b0302352.sam	6	-2.974	-5.782	M3
b0302352.sam	7	1.466	2.850	M3
b0302352.sam	8	1.449	2.817	M3
b0310252.sam	1	1.035	2.011	CW
b0310252.sam	2	1.059	2.058	CW
b0310252.sam	3	-3.299	-6.413	CW
b0310252.sam	4	-3.321	-6.456	CW
b0310252.sam	5	0.442	0.859	CW
b0310252.sam	6	3.321	6.455	CW
b0310252.sam	7	1.097	2.131	CW
b0310252.sam	8	1.093	2.124	CW
b0310552.sam	1	-0.379	-0.737	P
b0310552.sam	2	-0.413	-0.802	P
b0310552.sam	3	-0.020	-0.038	P
b0310552.sam	4	-0.302	-0.586	P
b0310552.sam	5	-2.232	-4.338	P
b0310552.sam	6	2.615	5.083	P
b0310552.sam	7	-0.024	-0.047	P
b0310552.sam	8	0.040	0.077	P
b0310852.sam	1	0.732	1.423	M1
b0310852.sam	2	0.683	1.328	M1
b0310852.sam	3	-3.236	-6.291	M1
b0310852.sam	4	0.711	1.383	M1
b0310852.sam	5	0.730	1.420	M1
b0310852.sam	6	-1.793	-3.486	M1
b0310852.sam	7	1.100	2.138	M1
b0310852.sam	8	1.091	2.120	M1
b0311152.sam	1	0.713	1.387	M4
b0311152.sam	2	0.702	1.365	M4
b0311152.sam	3	0.712	1.384	M4
b0311152.sam	4	0.759	1.475	M4
b0311152.sam	5	0.015	0.030	M4

Doppler estimates for Bermuda data set part 2.

file	channel	m/s	knots	signal
b0311152.sam	6	-2.911	-5.659	M4
b0311152.sam	7	0.772	1.500	M4
b0311152.sam	8	2.969	5.772	M4
b0311452.sam	1	1.392	2.706	CW
b0311452.sam	2	1.403	2.727	CW
b0311452.sam	3	1.447	2.814	CW
b0311452.sam	4	1.455	2.829	CW
b0311452.sam	5	-3.293	-6.401	CW
b0311452.sam	6	-3.260	-6.337	CW
b0311452.sam	7	1.374	2.670	CW
b0311452.sam	8	-0.377	-0.733	CW
b0311752.sam	1	2.174	4.226	P
b0311752.sam	2	2.095	4.073	P
b0311752.sam	3	-1.808	-3.514	P
b0311752.sam	4	-0.732	-1.422	P
b0311752.sam	5	-3.297	-6.408	P
b0311752.sam	6	-1.127	-2.191	P
b0311752.sam	7	1.923	3.738	P
b0311752.sam	8	3.329	6.471	P
b0312052.sam	1	1.453	2.825	M1
b0312052.sam	2	1.529	2.973	M1
b0312052.sam	3	3.365	6.541	M1
b0312052.sam	4	-3.285	-6.386	M1
b0312052.sam	5	2.926	5.689	M1
b0312052.sam	6	2.940	5.715	M1
b0312052.sam	7	1.520	2.954	M1
b0312052.sam	8	3.348	6.508	M1
b0312352.sam	1	1.839	3.574	M2
b0312352.sam	2	1.463	2.845	M2
b0312352.sam	3	-1.100	-2.138	M2
b0312352.sam	4	2.183	4.244	M2
b0312352.sam	5	1.098	2.134	M2
b0312352.sam	6	-1.162	-2.258	M2
b0312352.sam	7	1.502	2.920	M2
b0312352.sam	8	-0.406	-0.790	M2
b0320252.sam	1	-2.200	-4.277	NONE
b0320252.sam	2	1.556	3.024	NONE
b0320252.sam	3	-2.180	-4.237	NONE

Doppler estimates for Bermuda data set part 2.

file	channel	m/s	knots	signal
b0320252.sam	4	-3.293	-6.401	NONE
b0320252.sam	5	-1.458	-2.833	NONE
b0320252.sam	6	0.315	0.613	NONE
b0320252.sam	7	-3.263	-6.342	NONE
b0320252.sam	8	2.923	5.682	NONE
b0320552.sam	1	-2.561	-4.978	NONE
b0320552.sam	2	0.732	1.422	NONE
b0320552.sam	3	-1.160	-2.255	NONE
b0320552.sam	4	3.303	6.421	NONE
b0320552.sam	5	3.293	6.401	NONE
b0320552.sam	6	-3.282	-6.380	NONE
b0320552.sam	7	0.366	0.712	NONE
b0320552.sam	8	3.282	6.379	NONE
b0320852.sam	1	3.309	6.433	NONE
b0320852.sam	2	-3.286	-6.388	NONE
b0320852.sam	3	-3.283	-6.382	NONE
b0320852.sam	4	-3.279	-6.374	NONE
b0320852.sam	5	2.788	5.419	NONE
b0320852.sam	6	-1.508	-2.932	NONE
b0320852.sam	7	-0.483	-0.939	NONE
b0320852.sam	8	2.237	4.347	NONE
b0321152.sam	1	2.195	4.267	NONE
b0321152.sam	2	-0.768	-1.494	NONE
b0321152.sam	3	-2.195	-4.267	NONE
b0321152.sam	4	-2.170	-4.218	NONE
b0321152.sam	5	-3.276	-6.369	NONE
b0321152.sam	6	3.310	6.434	NONE
b0321152.sam	7	3.312	6.438	NONE
b0321152.sam	8	-0.732	-1.422	NONE

6.4 Capetown

The Capetown data set is contained on CD #2 in the directory **C.**.

Doppler estimates for Capetown data set.

file	channel	m/s	knots	signal
c0261540.sam	1	1.758	3.417	P
c0261845.sam	1	-2.934	-5.703	M1

Doppler estimates for Capetown data set.

file	channel	m/s	knots	signal
c0262140.sam	1	1.779	3.459	M4
c0271601.sam	1	1.502	2.919	P
c0271842.sam	1	-1.106	-2.150	NONE
c0271927.sam	1	-3.325	-6.462	NONE
c0281840.sam	1	-3.320	-6.453	NONE
c0290045.sam	1	0.055	0.107	NONE
c0290340.sam	1	-1.475	-2.867	NONE
c0290654.sam	1	0.709	1.378	NONE
c0290731.sam	1	-1.504	-2.923	NONE
c0290956.sam	1	0.400	0.777	M3
c0291240.sam	1	-0.344	-0.670	CW
c0291540.sam	1	-0.337	-0.655	P
c0291847.sam	1	-0.782	-1.520	M1
c0291912.sam	1	-0.716	-1.391	M1
c0292143.sam	1	0.626	1.217	M4
c0300041.sam	1	0.462	0.898	P
c0300642.sam	1	1.189	2.311	M1
c0300944.sam	1	1.065	2.071	M2
c0301240.sam	1	2.520	4.898	CW
c0301540.sam	1	1.449	2.817	P
c0301856.sam	1	1.172	2.279	M1
c0301945.sam	1	2.184	4.245	M1
c0302141.sam	1	0.689	1.340	M3
c0310041.sam	1	0.415	0.807	CW
c0310341.sam	1	-2.621	-5.095	P
c0310954.sam	1	-1.057	-2.054	M4
c0311240.sam	1	2.900	5.636	CW
c0311540.sam	1	0.744	1.447	P
c0311841.sam	1	0.327	0.636	M1
c0312140.sam	1	-2.522	-4.903	M2
c0320044.sam	1	-2.563	-4.981	NONE
c0320340.sam	1	-2.212	-4.299	NONE
c0320651.sam	1	-2.964	-5.762	NONE

6.5 Goa

Doppler estimates for Goa data set.

file	channel	m/s	knots	signal
g0240400.sam	1	3.275	6.367	NONE
g0241257.sam	1	-2.213	-4.302	NONE
g0241318.sam	1	-2.234	-4.342	NONE
g0241557.sam	1	-3.304	-6.423	NONE
g0241857.sam	1	2.570	4.995	NONE
g0250357.sam	1	3.305	6.425	NONE
g0250657.sam	1	1.473	2.863	NONE
g0250957.sam	1	-0.022	-0.043	NONE
g0251257.sam	1	-0.712	-1.385	NONE
g0251557.sam	1	-0.054	-0.105	NONE
g0251857.sam	1	-2.557	-4.970	NONE
g0252157.sam	1	2.572	4.999	NONE
g0260124.sam	1	1.612	3.133	CW
g0260357.sam	1	1.159	2.252	P
g0260657.sam	1	-3.061	-5.951	M1
g0260706.sam	1	-0.366	-0.711	M1
g0261627.sam	1	-1.097	-2.132	P
g0261637.sam	1	-1.056	-2.053	P
g0261857.sam	1	-1.398	-2.717	M1
g0262157.sam	1	-0.675	-1.312	M4
g0270057.sam	1	-0.654	-1.271	CW
g0271005.sam	1	-1.139	-2.213	M2
g0271044.sam	1	-1.148	-2.231	M2
g0271303.sam	1	-1.077	-2.094	CW
g0271557.sam	1	-0.963	-1.872	P
g0271857.sam	1	2.918	5.672	NONE
g0272157.sam	1	-3.287	-6.389	NONE
g0280057.sam	1	-2.927	-5.689	NONE
g0280405.sam	1	1.504	2.924	NONE
g0280453.sam	1	-0.321	-0.625	NONE
g0280657.sam	1	-1.493	-2.901	NONE
g0280957.sam	1	0.388	0.754	NONE
g0281302.sam	1	-0.414	-0.805	NONE
g0281609.sam	1	-3.331	-6.475	NONE
g0281857.sam	1	0.769	1.495	NONE
g0290421.sam	1	1.872	3.640	NONE
g0290436.sam	1	2.195	4.267	NONE
g0290657.sam	1	-2.181	-4.240	NONE
g0291036.sam	1	-1.950	-3.790	M3

Doppler estimates for Goa data set.

file	channel	m/s	knots	signal
g0291257.sam	1	-1.939	-3.769	CW
g0291557.sam	1	-1.948	-3.786	P
g0291857.sam	1	-1.900	-3.693	M1
g0300357.sam	1	0.783	1.523	P
g0300657.sam	1	0.476	0.925	M1
g0300957.sam	1	-0.357	-0.694	M2
g0301257.sam	1	-0.371	-0.722	CW
g0301557.sam	1	-0.349	-0.679	P
g0301857.sam	1	-0.262	-0.509	M1
g0320702.sam	1	2.591	5.036	NONE
g0331022.sam	1	0.746	1.451	NONE
g0331033.sam	1	1.867	3.630	NONE
g0331306.sam	1	-3.296	-6.408	NONE
g0331606.sam	1	2.600	5.054	NONE

6.6 Heard Island

Doppler estimates for Heard Island data set.

file	channel	m/s	knots	signal
h0290900.sam	1	-0.036	-0.070	M3
h0291156.sam	1	-1.439	-2.797	CW
h0291448.sam	1	-1.203	-2.338	P
h0291748.sam	1	-1.444	-2.807	M1
h0292048.sam	1	-2.141	-4.161	M4
h0292348.sam	1	-0.863	-1.678	CW
h0300248.sam	1	-1.625	-3.158	P
h0300548.sam	1	-1.511	-2.936	M1
h0300848.sam	1	-1.440	-2.799	M2
h0301148.sam	1	-1.913	-3.719	CW
h0301448.sam	1	-1.854	-3.604	P
h0310848.sam	1	-1.505	-2.926	M4

6.7 Kerguelen Island

Doppler estimates for Kerguelen Island data set.

file	channel	m/s	knots	signal
k0270302.sam	1	-0.262	-0.509	P
k0271500.sam	1	0.338	0.657	P
k0291200.sam	1	1.804	3.507	CW
k0302055.sam	1	-0.706	-1.372	M3
k0310000.sam	1	-0.881	-1.713	CW
k0310031.sam	1	-0.351	-0.683	CW

6.8 Mawson Station

Doppler estimates for Mawson Station data set.

file	channel	m/s	knots	signal
m0270607.sam	1	-2.961	-5.756	M1
m0271207.sam	1	1.445	2.809	CW
m0271507.sam	1	-1.496	-2.908	P
m0271807.sam	1	-3.334	-6.481	NONE
m0272107.sam	1	-0.395	-0.768	NONE
m0280317.sam	1	-2.498	-4.855	NONE
m0280907.sam	1	1.942	3.775	NONE
m0281207.sam	1	3.357	6.525	NONE
m0281507.sam	1	1.475	2.867	NONE
m0281807.sam	1	2.902	5.641	NONE
m0282107.sam	1	-3.293	-6.400	NONE
m0290007.sam	1	0.700	1.361	NONE
m0290307.sam	1	-2.182	-4.241	NONE
m0290607.sam	1	-0.345	-0.670	NONE
m0300607.sam	1	-0.461	-0.896	M1
m0300907.sam	1	-3.275	-6.367	M2
m0301524.sam	1	-2.928	-5.692	P
m0301807.sam	1	3.321	6.456	M1
m0302107.sam	1	2.595	5.044	M3
m0310007.sam	1	-2.890	-5.617	CW
m0310308.sam	1	-0.615	-1.195	P
m0310612.sam	1	1.480	2.877	M1
m0310907.sam	1	2.187	4.252	M4
m0311207.sam	1	2.959	5.752	CW
m0311507.sam	1	1.823	3.543	P
m0311807.sam	1	-1.811	-3.520	M1

Doppler estimates for Mawson Station data set.

file	channel	m/s	knots	signal
m0312107.sam	1	-0.009	-0.018	M2

6.9 Russia (Krylov Seamount)

This data was received split out into “seances” and individual A/D channels. We did not receive the original “R0xxxxxx” files. The single seance data was reformatted into the basic same format used by the standard HIFT CompuAdd systems. No attempt was made to recombine the single channel sets into a multichannel format. The following table was also supplied with the data for use in mapping between original data files and the seance designations. The S letter designation for this data derives from the use of the word seance.

Seance	Date	Time	Name	Signal
1	Jan 26, 1991	0210	R0260210	CW
2	Jan 26, 1991	0510	R0260510	P
3	Jan 26, 1991	0810	R0260810	M1
4	Jan 26, 1991	1110	R0261110	M3
5	Jan 26, 1991	1410	R0261410	CW
6	Jan 26, 1991	1710	R0261710	P
7	Jan 26, 1991	2010	R0262110	M1
8	Jan 26, 1991	2310	R0262310	M4
9	Jan 27, 1991	0210	R0270210	CW
10	Jan 27, 1991	0510	R0270510	P
11	Jan 27, 1991	0810	R0270810	M1
12	Jan 27, 1991	1110	R0271110	M2
13	Jan 27, 1991	1410	R0271410	CW
14	Jan 27, 1991	1710	R0271710	P
28	Jan 29, 1991	1110	R0291110	M3
29	Jan 29, 1991	1410	R0291410	CW
30	Jan 29, 1991	1710	R0291710	P
31	Jan 29, 1991	2010	R0292010	M1
32	Jan 29, 1991	2310	R0292310	M4
33	Jan 30, 1991	0210	R0300210	CW
34	Jan 30, 1991	0510	R0300510	P
35	Jan 30, 1991	0810	R0300810	M1
36	Jan 30, 1991	1110	R0301110	M2
37	Jan 30, 1991	1410	R0301410	CW
38	Jan 30, 1991	1710	R0301710	P
39	Jan 30, 1991	2010	R0302010	M1
40	Jan 30, 1991	2310	R0302310	M3
41	Jan 31, 1991	0210	R0310210	CW
42	Jan 31, 1991	0510	R0310510	P
43	Jan 31, 1991	0810	R0310810	M1
44	Jan 31, 1991	1110	R0311110	M4
45	Jan 31, 1991	1410	R0311410	CW
46	Jan 31, 1991	1710	R0311710	P
47	Jan 31, 1991	2010	R0312010	M1
48	Jan 31, 1991	2310	R0312310	M2

Doppler estimates for Krylov Seamount data set.

file	channel	m/s	knots	From	Signal
s01_3.sam	1	-0.306	-0.595	R0260210	CW
s01_4.sam	1	-0.252	-0.491	R0260210	CW
s02_3.sam	1	0.993	1.930	R0260510	P
s02_4.sam	1	0.889	1.728	R0260510	P
s03_2.sam	1	1.491	2.898	R0260810	M1
s03_3.sam	1	1.488	2.892	R0260810	M1
s04_1.sam	1	1.512	2.938	R0261110	M3
s04_2.sam	1	1.517	2.949	R0261110	M3
s04_3.sam	1	1.503	2.922	R0261110	M3
s05_1.sam	1	1.580	3.072	R0261410	CW
s05_2.sam	1	1.581	3.074	R0261410	CW
s05_3.sam	1	1.538	2.990	R0261410	CW
s06_3.sam	1	1.357	2.638	R0261710	P
s06_4.sam	1	1.320	2.566	R0261710	P
s07_1.sam	1	1.335	2.595	R0262110	M1
s07_3.sam	1	1.167	2.269	R0262110	M1
s07_4.sam	1	1.332	2.590	R0262110	M1
s08_1.sam	1	1.377	2.677	R0262310	M4
s08_3.sam	1	1.404	2.729	R0262310	M4
s08_4.sam	1	1.394	2.709	R0262310	M4
s09_1.sam	1	1.245	2.420	R0270210	CW
s09_3.sam	1	1.215	2.361	R0270210	CW
s09_4.sam	1	1.242	2.415	R0270210	CW
s10_1.sam	1	1.838	3.573	R0270510	P
s10_3.sam	1	1.825	3.548	R0270510	P
s10_4.sam	1	1.830	3.557	R0270510	P
s11_1.sam	1	1.490	2.897	R0270810	M1
s11_3.sam	1	1.415	2.750	R0270810	M1
s11_4.sam	1	1.482	2.881	R0270810	M1
s12_1.sam	1	1.804	3.507	R0271110	M2
s12_3.sam	1	1.768	3.438	R0271110	M2
s12_4.sam	1	1.784	3.467	R0271110	M2
s13_3.sam	1	1.485	2.886	R0271410	CW
s13_4.sam	1	1.507	2.929	R0271410	CW
s14_3.sam	1	1.118	2.174	R0271710	P

Doppler estimates for Krylov Seamount data set.

file	channel	m/s	knots	From	Signal
s14_4.sam	1	1.272	2.472	R0271710	P
s28_4.sam	1	-0.228	-0.442	R0291110	M3
s29_4.sam	1	-0.413	-0.804	R0291410	CW
s30_4.sam	1	-0.021	-0.041	R0291710	P
s31_4.sam	1	0.049	0.096	R0292010	M1
s32_4.sam	1	0.952	1.851	R0292310	M4
s33_4.sam	1	0.414	0.805	R0300210	CW
s34_4.sam	1	1.083	2.105	R0300510	P
s35_4.sam	1	1.357	2.638	R0300810	M1
s36_4.sam	1	1.507	2.929	R0301110	M2
s37_4.sam	1	2.101	4.084	R0301410	CW
s38_4.sam	1	1.686	3.276	R0301710	P
s39_3.sam	1	1.790	3.480	R0302010	M1
s39_4.sam	1	1.715	3.333	R0302010	M1
s40_3.sam	1	1.142	2.219	R0302310	M3
s40_4.sam	1	1.181	2.296	R0302310	M3
s41_3.sam	1	1.010	1.963	R0310210	CW
s41_4.sam	1	0.981	1.908	R0310210	CW
s42_3.sam	1	-1.502	-2.919	R0310510	P
s42_4.sam	1	-0.314	-0.610	R0310510	P
s43_3.sam	1	1.037	2.015	R0310810	M1
s43_4.sam	1	0.754	1.466	R0310810	M1
s44_4.sam	1	0.449	0.873	R0311110	M4
s45_4.sam	1	1.188	2.308	R0311410	CW
s46_4.sam	1	1.876	3.646	R0311710	P
s47_4.sam	1	1.181	2.296	R0312010	M1
s48_4.sam	1	1.464	2.845	R0312310	M2
s49_4.sam	1	-1.815	-3.527	???	NONE?
s73_4.sam	1	-3.293	-6.401	???	NONE?

6.10 Tasmania

Doppler estimates for Tasmania data set.

file	channel	m/s	knots	signal
t0290051.sam	1	1.829	3.556	NONE
t0290346.sam	1	-2.195	-4.267	NONE
t0290645.sam	1	-2.195	-4.267	NONE
t0290945.sam	1	0.732	1.422	M3
t0291245.sam	1	-3.293	-6.401	CW
t0291545.sam	1	-1.432	-2.784	P
t0291846.sam	1	2.916	5.669	M1
t0292145.sam	1	3.278	6.372	M4
t0300107.sam	1	0.732	1.422	CW
t0300345.sam	1	-3.304	-6.422	P
t0300645.sam	1	1.866	3.626	M1
t0300946.sam	1	-2.174	-4.226	M2
t0301245.sam	1	2.937	5.709	CW
t0301545.sam	1	0.419	0.815	P
t0301850.sam	1	-0.419	-0.814	M1
t0302146.sam	1	2.195	4.267	M3
t0310046.sam	1	-1.824	-3.546	CW
t0310346.sam	1	-1.468	-2.853	P
t0310645.sam	1	1.799	3.496	M1
t0310945.sam	1	3.267	6.351	M4
t0311245.sam	1	0.347	0.674	CW
t0311548.sam	1	2.927	5.689	P
t0311857.sam	1	-3.294	-6.403	M1
t0312147.sam	1	2.174	4.227	M2
t0320055.sam	1	3.299	6.413	NONE

6.11 West Coast of U.S.

Doppler estimates for West Coast of U.S. data set.

file	channel	m/s	knots	signal
w0260316.sam	1	-3.351	-6.513	CW
w0260316.sam	2	3.293	6.401	CW
w0260316.sam	3	-3.336	-6.484	CW
w0260316.sam	4	-0.782	-1.520	CW
w0260316.sam	5	-0.768	-1.493	CW
w0260316.sam	6	-0.681	-1.325	CW
w0260316.sam	7	-0.759	-1.476	CW

Doppler estimates for West Coast of U.S. data set.

file	channel	m/s	knots	signal
w0260316.sam	8	-1.053	-2.047	CW
w0260616.sam	1	-1.466	-2.849	P
w0260616.sam	2	-2.578	-5.010	P
w0260616.sam	3	-2.512	-4.883	P
w0260616.sam	4	-1.475	-2.866	P
w0260616.sam	5	-0.363	-0.705	P
w0260616.sam	6	3.377	6.563	P
w0260616.sam	7	-3.336	-6.484	P
w0260616.sam	8	-0.728	-1.416	P
w0260916.sam	1	1.066	2.072	M1
w0260916.sam	2	-2.571	-4.997	M1
w0260916.sam	3	2.561	4.978	M1
w0260916.sam	4	-1.780	-3.460	M1
w0260916.sam	5	-1.524	-2.962	M1
w0260916.sam	6	-0.719	-1.397	M1
w0260916.sam	7	-2.552	-4.961	M1
w0260916.sam	8	-2.931	-5.698	M1
w0261216.sam	1	1.091	2.122	M3
w0261216.sam	2	-1.076	-2.091	M3
w0261216.sam	3	-1.099	-2.137	M3
w0261216.sam	4	-1.518	-2.952	M3
w0261216.sam	5	-1.506	-2.927	M3
w0261216.sam	6	-0.785	-1.525	M3
w0261216.sam	7	-2.941	-5.716	M3
w0261216.sam	8	-3.291	-6.396	M3
w0261516.sam	1	-3.292	-6.398	CW
w0261516.sam	2	3.329	6.470	CW
w0261516.sam	3	2.893	5.623	CW
w0261516.sam	4	-2.879	-5.596	CW
w0261516.sam	5	-0.774	-1.505	CW
w0261516.sam	6	2.968	5.769	CW
w0261516.sam	7	-3.288	-6.392	CW
w0261516.sam	8	-2.211	-4.297	CW
w0261816.sam	1	-3.282	-6.379	P
w0261816.sam	2	-3.347	-6.506	P
w0261816.sam	3	3.365	6.540	P
w0261816.sam	4	-0.364	-0.708	P
w0261816.sam	5	-1.087	-2.114	P

Doppler estimates for West Coast of U.S. data set.

file	channel	m/s	knots	signal
w0261816.sam	6	-1.459	-2.836	P
w0261816.sam	7	-0.367	-0.713	P
w0261816.sam	8	-2.219	-4.314	P
w0262116.sam	1	-0.365	-0.710	M1
w0262116.sam	2	-0.024	-0.046	M1
w0262116.sam	3	2.195	4.266	M1
w0262116.sam	4	-0.285	-0.554	M1
w0262116.sam	5	1.829	3.556	M1
w0270055.sam	1	-0.746	-1.451	M4
w0270055.sam	2	-0.689	-1.340	M4
w0270055.sam	3	2.220	4.315	M4
w0270055.sam	4	3.293	6.401	M4
w0270055.sam	5	-2.895	-5.628	M4
w0270307.sam	1	-0.771	-1.499	CW
w0270307.sam	2	-0.769	-1.495	CW
w0270307.sam	3	-0.664	-1.291	CW
w0270307.sam	4	-1.480	-2.876	CW
w0270607.sam	1	-0.749	-1.456	P
w0270607.sam	2	-0.743	-1.445	P
w0270607.sam	3	-0.728	-1.415	P
w0270607.sam	4	-0.755	-1.469	P
w0270907.sam	1	-0.384	-0.746	M1
w0270907.sam	2	3.236	6.291	M1
w0270907.sam	3	-0.435	-0.846	M1
w0270907.sam	4	-1.077	-2.094	M1
w0271207.sam	1	-0.379	-0.737	M2
w0271207.sam	2	-0.655	-1.274	M2
w0271207.sam	3	-0.459	-0.891	M2
w0271207.sam	4	-0.691	-1.342	M2
w0271507.sam	1	-0.361	-0.702	CW
w0271507.sam	2	-0.410	-0.796	CW
w0271507.sam	3	-0.366	-0.711	CW
w0271507.sam	4	-0.355	-0.689	CW
w0271807.sam	1	-0.349	-0.679	P
w0271807.sam	2	3.290	6.395	P
w0271807.sam	3	-0.380	-0.739	P

Doppler estimates for West Coast of U.S. data set.

file	channel	m/s	knots	signal
w0271807.sam	4	-1.463	-2.843	P
w0280007.sam	1	0.391	0.761	NONE
w0280007.sam	2	3.228	6.274	NONE
w0280007.sam	3	1.106	2.150	NONE
w0280007.sam	4	-0.706	-1.372	NONE
w0280307.sam	1	3.293	6.401	NONE
w0280307.sam	2	-0.366	-0.711	NONE
w0280307.sam	3	1.812	3.522	NONE
w0280307.sam	4	-0.758	-1.473	NONE
w0280607.sam	1	3.288	6.391	NONE
w0280607.sam	2	2.288	4.448	NONE
w0280607.sam	3	-3.355	-6.522	NONE
w0280607.sam	4	3.342	6.495	NONE
w0280907.sam	1	2.561	4.978	NONE
w0280907.sam	2	2.195	4.267	NONE
w0280907.sam	3	-3.260	-6.336	NONE
w0280907.sam	4	-2.954	-5.741	NONE
w0281207.sam	1	-0.381	-0.741	NONE
w0281207.sam	2	3.340	6.492	NONE
w0281207.sam	3	-2.880	-5.598	NONE
w0281207.sam	4	1.486	2.888	NONE
w0281507.sam	1	2.213	4.302	NONE
w0281507.sam	2	1.805	3.509	NONE
w0281507.sam	3	1.738	3.378	NONE
w0281507.sam	4	-0.048	-0.093	NONE
w0281807.sam	1	-3.280	-6.376	NONE
w0281807.sam	2	2.945	5.725	NONE
w0281807.sam	3	2.225	4.326	NONE
w0281807.sam	4	-2.907	-5.651	NONE
w0290907.sam	1	-1.149	-2.234	NONE
w0290907.sam	2	-3.325	-6.464	NONE
w0290907.sam	3	-3.280	-6.376	NONE
w0290907.sam	4	-2.147	-4.173	NONE
w0291207.sam	1	1.166	2.266	M3
w0291207.sam	2	1.120	2.177	M3
w0291207.sam	3	1.136	2.208	M3

Doppler estimates for West Coast of U.S. data set.

file	channel	m/s	knots	signal
w0291207.sam	4	2.949	5.732	M3
w0291507.sam	1	1.429	2.778	CW
w0291507.sam	2	1.381	2.684	CW
w0291507.sam	3	1.408	2.737	CW
w0291507.sam	4	1.442	2.804	CW
w0291807.sam	1	1.141	2.219	P
w0291807.sam	2	1.072	2.083	P
w0291807.sam	3	1.304	2.535	P
w0291807.sam	4	-2.907	-5.650	P
w0292107.sam	1	1.110	2.158	M1
w0292107.sam	2	1.128	2.194	M1
w0292107.sam	3	-3.296	-6.407	M1
w0292107.sam	4	-3.284	-6.384	M1
w0300007.sam	1	-1.878	-3.650	M4
w0300007.sam	2	-1.858	-3.611	M4
w0300007.sam	3	-3.313	-6.440	M4
w0300007.sam	4	-3.285	-6.385	M4
w0300307.sam	1	-1.018	-1.979	CW
w0300307.sam	2	-0.991	-1.926	CW
w0300307.sam	3	-1.075	-2.090	CW
w0300307.sam	4	-1.034	-2.010	CW
w0300607.sam	1	-1.502	-2.920	P
w0300607.sam	2	-1.480	-2.876	P
w0300607.sam	3	-1.520	-2.955	P
w0300607.sam	4	-1.525	-2.965	P
w0300907.sam	1	-1.588	-3.088	M1
w0300907.sam	2	-1.559	-3.030	M1
w0300907.sam	3	-1.482	-2.881	M1
w0300907.sam	4	-1.541	-2.996	M1
w0301207.sam	1	-1.223	-2.377	M2
w0301207.sam	2	-1.245	-2.419	M2
w0301207.sam	3	-1.450	-2.819	M2
w0301207.sam	4	-1.031	-2.004	M2
w0301507.sam	1	-1.544	-3.002	CW
w0301507.sam	2	-1.572	-3.055	CW
w0301507.sam	3	-1.542	-2.998	CW

Doppler estimates for West Coast of U.S. data set.

file	channel	m/s	knots	signal
w0301507.sam	4	-1.508	-2.932	CW
w0301807.sam	1	-1.147	-2.229	P
w0301807.sam	2	-1.112	-2.161	P
w0301807.sam	3	-1.435	-2.790	P
w0301807.sam	4	-1.121	-2.180	P
w0302107.sam	1	-1.374	-2.671	M1
w0302107.sam	2	-1.439	-2.797	M1
w0302107.sam	3	-1.425	-2.770	M1
w0302107.sam	4	3.262	6.342	M1
w0310007.sam	1	-1.217	-2.365	M3
w0310007.sam	2	-1.112	-2.162	M3
w0310007.sam	3	3.333	6.479	M3
w0310007.sam	4	-1.106	-2.149	M3
w0310307.sam	1	-1.048	-2.037	CW
w0310307.sam	2	2.567	4.989	CW
w0310307.sam	3	2.486	4.833	CW
w0310307.sam	4	-2.930	-5.695	CW
w0310607.sam	1	-0.725	-1.409	P
w0310607.sam	2	-1.123	-2.183	P
w0310607.sam	3	-2.563	-4.982	P
w0310607.sam	4	-2.606	-5.066	P
w0310607.sam	5	-0.052	-0.100	P
w0310907.sam	1	-1.843	-3.582	M1
w0310907.sam	2	-1.102	-2.142	M1
w0310907.sam	3	-1.451	-2.820	M1
w0310907.sam	4	-3.294	-6.404	M1
w0310907.sam	5	-0.018	-0.035	M1
w0311207.sam	1	-1.498	-2.912	M4
w0311207.sam	2	-1.522	-2.958	M4
w0311207.sam	3	-1.863	-3.621	M4
w0311207.sam	4	1.128	2.193	M4
w0311207.sam	5	0.022	0.042	M4
w0311507.sam	1	-3.321	-6.455	CW
w0311507.sam	2	1.107	2.151	CW
w0311507.sam	3	-2.109	-4.100	CW
w0311507.sam	4	-1.436	-2.791	CW

Doppler estimates for West Coast of U.S. data set.

file	channel	m/s	knots	signal
w0311507.sam	5	-1.057	-2.055	CW
w0311807.sam	1	-1.483	-2.883	P
w0311807.sam	2	-2.210	-4.297	P
w0311807.sam	3	1.583	3.077	P
w0311807.sam	4	-2.521	-4.900	P
w0311807.sam	5	3.293	6.401	P
w0312107.sam	1	-0.379	-0.737	M1
w0312107.sam	2	3.310	6.435	M1
w0312107.sam	3	0.323	0.628	M1
w0312107.sam	4	3.355	6.521	M1
w0312107.sam	5	-0.699	-1.359	M1
w0320007.sam	1	-2.175	-4.227	M2
w0320007.sam	2	-3.241	-6.299	M2
w0320007.sam	3	-2.163	-4.205	M2
w0320007.sam	4	-1.098	-2.134	M2
w0320007.sam	5	-3.195	-6.212	M2
w0320307.sam	1	-3.261	-6.338	NONE
w0320307.sam	2	3.317	6.447	NONE
w0320307.sam	3	-1.814	-3.525	NONE
w0320307.sam	4	3.323	6.459	NONE
w0320307.sam	5	1.994	3.877	NONE

6.12 Christmas (Xmas) Island

Doppler estimates for Christmas Island data set.

file	channel	m/s	knots	signal
x0260954.sam	1	-0.378	-0.734	M3
x0261249.sam	1	-1.461	-2.841	CW
x0261306.sam	1	-1.378	-2.678	CW
x0261354.sam	1	-0.375	-0.729	CW
x0261549.sam	1	-1.487	-2.891	P
x0261849.sam	1	-1.627	-3.163	M1
x0262149.sam	1	-1.199	-2.330	M4
x0270049.sam	1	-1.094	-2.126	CW
x0270349.sam	1	-1.807	-3.513	P
x0270649.sam	1	-1.476	-2.868	M1
x0270949.sam	1	-1.798	-3.495	M2

Doppler estimates for Christmas Island data set.

file	channel	m/s	knots	signal
x0271249.sam	1	-1.676	-3.259	CW
x0271549.sam	1	-1.475	-2.866	P
x0271849.sam	1	2.222	4.319	NONE
x0272149.sam	1	0.371	0.721	NONE
x0280049.sam	1	3.278	6.372	NONE
x0280138.sam	1	-1.491	-2.898	NONE
x0280349.sam	1	-2.927	-5.690	NONE
x0280357.sam	1	-1.464	-2.845	NONE
x0280401.sam	1	-1.445	-2.809	NONE
x0280649.sam	1	2.588	5.030	NONE
x0280651.sam	1	-1.480	-2.878	NONE
x0280949.sam	1	-3.294	-6.404	NONE
x0281249.sam	1	2.887	5.611	NONE
x0281254.sam	1	3.299	6.412	NONE
x0281607.sam	1	-1.132	-2.200	NONE
x0290649.sam	1	1.410	2.741	NONE
x0290949.sam	1	-1.446	-2.810	M3
x0291249.sam	1	3.355	6.522	CW
x0291549.sam	1	-1.324	-2.574	P
x0291849.sam	1	-1.329	-2.583	M1
x0292149.sam	1	0.665	1.293	M4
x0300049.sam	1	0.477	0.928	CW
x0300349.sam	1	0.116	0.226	P
x0300649.sam	1	-0.288	-0.560	M1
x0300949.sam	1	-1.136	-2.208	M2
x1280349.sam	1	2.927	5.689	NONE

6.13 New Zealand

Doppler estimates for New Zealand data set.

file	channel	m/s	knots	signal
z0300702.sam	1	-2.043	-3.970	M1
z0302204.sam	1	-1.439	-2.797	M3
z0310404.sam	1	-0.789	-1.533	P
z0310704.sam	1	-1.859	-3.614	M1
z0311004.sam	1	-1.474	-2.864	M4
z0320407.sam	1	-0.006	-0.012	NONE

z0320707.sam	1	-0.809	-1.573	NONE

A atoccrc source code

```
/* file name: atoccrc.c

   Program to form CRC for input file.  To be used to check for errors in
   large file transfers on the internet.  Compute the CRCs of the file
   both before and after transfer and compare.

   atoccrc file_with_list_of_file_names

   Output is to stdout and can be redirected to a file.

   13Dec96..initial version..KM
   18Dec96..converted from 16 bit to 32 bit CRC..KM
   23Sep97..accepts input file name list..KM
*/

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

#define MAX_NAME_SIZE 100

char file_name[MAX_NAME_SIZE+1];

int main(int argc, char** argv)
{
  int i, count, file_index;
  unsigned long crc, crc_bit, byte, byte_bit;
  FILE *input, *list;

  count=0;
  file_index=1;
  if (argc!=2) {
    printf("improper number of args\n");
    exit(1);
  }
  if ((list=fopen(argv[1], "r"))==NULL) {
    printf("can't open file list file %s\n", argv[1]);
    exit(1);
  }

  while (!feof(list)) {
    if (fgets(file_name, MAX_NAME_SIZE, list)==NULL) continue;
if (feof(list)) continue;

for (i=0; file_name[i]!='\0'; i++)
  if ((file_name[i]=='\n')||(file_name[i]=='\r')) file_name[i]='\0';
```

```
    if ((input=fopen(file_name, "rb"))==NULL) {
      fprintf(stderr, "can't open input file %s\n", file_name);
      continue;
    }
    fprintf(stderr, "file: %3d  %s\n", file_index, file_name);
    crc=0;
    while (!feof(input)) {
      byte=(unsigned long)fgetc(input);
      if (feof(input)) continue;
      for (i=0; i<8; i++) {
        crc_bit=crc&1;
        crc>>=1;
        byte_bit=byte&1;
        byte>>=1;
        crc_bit^=byte_bit;
        if (crc_bit) {
          crc^=0xEDB88320;
        }
      }
    }
    fclose(input);

/* strip path off of file name */

    i=strlen(file_name);
    while (i>0) {
      i--;
      if ((file_name[i]=='\\')||(file_name[i]==':')) {i++; break;}
    }
    printf("%-14s  %08lX\n", &file_name[i], crc);
    count++;
    file_index++;
  }
  printf("files processed: %d\n", count);
  fclose(list);
  return (0);
}
```

B crctest source code

```
// CRCtest.cpp :
//
//  Program to generate and check CRC values in current directory (.) and
//    associated subdirectories.
//
//  Usage:
//
//  crctest         Starts with the current directory (.) and generates
```

```
//                  the CRCs for all files in this directory and all
//                  subdirectories.  Values go to stdout.
//
//  crctest f       Start at current path verifiy CRCs listed in f.
//                  .
//  crctest p f     Start at directory p and verify CRCs listed in f.
//
//  Lines in the input file starting with a space are assumed are to contain
//  checksum in hex followed by file name including path.  Lines not starting
//  with a space are assumed to be other information and are skipped over.

//  Modeled after "File Verification Using CRC" by Mark Nelson
//  in Dr. Dobb's Journal May, 1992.  Also available on the web.
//
//  Table lookup CRC generation modeled after code by Gary S. Brown.
//
//  19June2000..Initial MSWindows only version for HIFT use..K.Metzger
//  21June2000..Changed initial CRC value for non h option use..K.Metzger
//  27June2000..UNIX (HP) support added..K.Metzger
//
//  To do:
//     Handle spaces in user supplied path and file designations.
//

#include <stdio.h>
#include <time.h>

#ifdef WIN32
#include <windows.h>
char SL='\\';
char SLS[]="\\";
#else
#include <dirent.h>
#include <sys/types.h>
#include <sys/stat.h>
char SL='/';
char SLS[]="/";
#endif

void MakeCRCs(char *);
void CheckCRCs(char *);
void CalcCRCtable(void);
unsigned long CalcCRC(char *);

#define BufferSize 256

char cID[]="CRCtest(27June2000)";
char cCRCFileName[BufferSize+1];
char cBasePath[BufferSize+1];
char cTemp[BufferSize+1];
char cMode[BufferSize+1];
```

```
char cCRCMode;
char cCase;
char *cpT;
char cT;
long lDirCount, lFileCount;
unsigned long ulCRCtime;
FILE *output;

int main(int argc, char* argv[])
{
    char *myargv[4];
    clock_t tStart, tEnd, tDate;
    int i, iCtr;
    double dTimeDifference;

    tDate = time(NULL);
    printf("%s executed at %s", cID, ctime(&tDate));
    output = stdout;
    strcpy(cMode, "verify");
    strcpy(cBasePath, ".");
    CalcCRCtable();
    cCRCMode = ' ';
    cCase = ' ';

    iCtr = 0;
    for (i = 0; i < argc; i++) {
        strcpy(cTemp, argv[i]);
        if (strcmp(cTemp, "-H") == 0) {
            cCRCMode = 'h';  // use 0 as initial CRC value
            continue;
        }
        if (strcmp(cTemp, "-h") == 0) {
            cCRCMode = 'h';
            strcpy(cMode, "hift"); // set up old HIFT compatabtily
            continue;
        }
        if (strcmp(cTemp, "-g") == 0) {
            strcpy(cMode, "generate"); // genenerate CRC file
            continue;
        }
        if (strcmp(cTemp, "-l") == 0) {
            cCase = 'l';
            continue;
        }
        if (strcmp(cTemp, "-u") == 0) {
            cCase = 'u';
            continue;
        }

        myargv[iCtr] = argv[i];
        iCtr++;
```

```
}
argc = iCtr;


// If not 2 or 3 args then give the user help
//
if ((argc == 1) || (argc > 3)) {
    printf("usage:\n");
    printf("  crctest file\n");
    printf("     base_path = . and crcs in file\n");
    printf("  crctest base_path crc_file\n");
    printf("     checks CRC info based on crc_file contents\n");
    printf("files are relative to the base directory\n");
    if (argc > 3) exit(1);
}

// Only get here is 1, 2, or 3 arg values,
// These are the program name and one or two additional.
//
if (argc == 2) {
    strcpy(cBasePath, ".");
    strcpy(cCRCFileName, myargv[1]);
}
else if (argc == 3) {
    strcpy(cBasePath, myargv[1]);
    strcpy(cCRCFileName, myargv[2]);
}

// only get here if no program args given..interact!
//
else {

    // Get and check validty of operating mode.
    //
    printf("\n");
    printf("mode (generate, verify, hift): ");
    scanf("%s", cMode);
    fgetc(stdin); // flush input buffer
    if (!( (*cMode=='g')||(*cMode=='v')||(*cMode=='h') )) {
        printf("invalid mode\n");
        exit(1);
    }

    // Get the base path to use for data files.
    //
    printf("enter base path: ");
    fgets(cTemp, BufferSize, stdin);
    if (strlen(cTemp) !=0) {
        sscanf(cTemp, "%s", cBasePath);
    }
```

```
        // Get the location and name of the CRC file
        //
        do {
            printf("crc file: ");
            fgets(cTemp, BufferSize, stdin);
            sscanf(cTemp, "%s", cCRCFileName);
            if ((strlen(cCRCFileName) == 0) && (cMode[0] ='g')) break;
            if (strlen(cCRCFileName) != 0) break;
            printf("must enter crc file name!\n");
        } while (1);
    }

    // If generating CRCs open the file for writing.
    // If no name just use stdout.
    //
    if ((strlen(cCRCFileName) != 0) && (cMode[0] == 'g')) {
        if ((output = fopen(cCRCFileName, "wt")) == NULL) {
            printf("can't open %s\n", cCRCFileName);
            exit(1);
        }
        fprintf(output, "%s executed at %s", cID, ctime(&tDate));
    }

    // If the base is dir: then don't append a \
    // If the base is a name without an ending \ then add a ending \
    //
    cT = SL;             // proper slash character
    strcpy(cTemp, SLS);  // proper slash string
    if ((cBasePath[strlen(cBasePath)-1] != ':' ) &&
        (cBasePath[strlen(cBasePath)-1] != cT)) {
        strcat(cBasePath, cTemp);
    }

    sprintf(cTemp, "base directory: %s\n", cBasePath);
    if (output != stdout) fprintf(output, cTemp);
    printf(cTemp);

    // If generating CRCs use this code.
    //
    if (*cMode == 'g') {
        printf("executing CRC generation\n");
        lFileCount = 0;
        lDirCount = 0;
        tStart = clock();
        MakeCRCs("");
        tEnd = clock();
        sprintf(cTemp, "%ld files in %ld directories\n", lFileCount, lDirCount);
        fprintf(output, cTemp);
        if (output != stdout) printf(cTemp);
    }
```

```
    // If checking CRCs then use this code.
    //
    else {
        if (*cMode == 'h') cCRCMode = 'h';
        printf("executing CRC verification\n");
        lFileCount = 0;
        tStart = clock();
        CheckCRCs(cCRCFileName);
        tEnd = clock();
        printf("%ld file CRCs checked\n", lFileCount);
    }

    dTimeDifference = (double)(tEnd-tStart)/CLOCKS_PER_SEC;
    sprintf(cTemp, "execution time: %.3lf seconds\n", dTimeDifference);
    if (output != stdout) {
        fprintf(output, cTemp);
        fclose(output);
    }
    printf(cTemp);

    return 0;
}

// Generate CRCs for all files in starting directory and
// all subdirectories.
//
void MakeCRCs(char* cDName)
{

#ifdef WIN32
#define NAME FindFileData.cFileName
    HANDLE hFile;
    WIN32_FIND_DATA FindFileData;
#else
#define INVALID_HANDLE_VALUE NULL
#define NAME Entry->d_name
#define STAT stat
    DIR *hFile;
    struct dirent *Entry;
    struct stat EntryStatus;
#endif
    char cRelPath[BufferSize+1];
    int iFound;
    unsigned long crc;

    lDirCount++;

    strcpy(cRelPath, cDName);
    if (strlen(cRelPath) != 0) strcat(cRelPath, SLS);
    strcpy(cTemp, cBasePath);
    strcat(cTemp, cRelPath);
```

```
#ifdef WIN32
    strcat(cTemp, "*");
    hFile = FindFirstFile(cTemp, &FindFileData);
    iFound = hFile != NULL;
#else
    hFile = opendir(cTemp);
    if (hFile == INVALID_HANDLE_VALUE) {
        printf("can't open directory: %s\n", cTemp);
        exit(1);
    }
    Entry = readdir(hFile);
    iFound = Entry != NULL;
#endif

    while (iFound != 0) {
        if ((strcmp(NAME, ".")!=0) &&
            (strcmp(NAME, "..")!=0)) {

            strcpy(cTemp, cBasePath);
            strcat(cTemp, cRelPath);
            strcat(cTemp, NAME);
#ifdef WIN32
            if (FindFileData.dwFileAttributes & FILE_ATTRIBUTE_DIRECTORY) {
#else
            if (STAT(cTemp, &EntryStatus) != 0) {
                printf("status was not found for %s\n", cTemp);
                exit(1);
            }
            if (EntryStatus.st_mode & S_IFDIR) {
#endif
                strcpy(cTemp, cRelPath);
                strcat(cTemp, NAME);
                MakeCRCs(cTemp);
            }
            else {
                lFileCount++;
                crc = CalcCRC(cTemp);
                strcpy(cTemp, cRelPath);
                strcat(cTemp, NAME);
                fprintf(output, " %08lX .\\%s\n", crc, cTemp);
            }
        }
#ifdef WIN32
        iFound = FindNextFile(hFile, &FindFileData) != NULL;
#else
        Entry = readdir(hFile);
        iFound = Entry != NULL;
#endif
    }

    // What was acquired must be disposed of!
```

```
    //
#ifdef WIN32
    if (FindClose(hFile) == 0) {
#else
    if (closedir(hFile) != 0) {
#endif
        printf("could not release handle for %s\n", cRelPath);
        exit(1);
    }


    return;

    printf("CRC generation not supported\n");
    exit(1);
}

// Compute table for CRC calcualation on a byte basis.
//
// X^32+X^26+X^23+X^22+X^16+X^12+X^11+X^10+X^8+X^7+X^5+X^4+X^2+X^1+X^0
//
// Write this in binary as a 33 bit number.  Drop the bit corresponding
// to X^32.  Reverse the bit ordering.  For the above polynomial the
// result is: 0xEDB88320.
//
static unsigned long crc_32_tab[256]; // used for CRC table lookup

void CalcCRCtable(void)
{
    unsigned long crc, byte_bit, crc_bit, byte, i, val;

    for (val = 0; val < 256; val++) {
        byte = val;
        crc = 0;
        for (i=0; i<8; i++) {
            crc_bit=crc&1;
            crc>>=1;
            byte_bit=byte&1;
            byte>>=1;
            crc_bit^=byte_bit;
            if (crc_bit) {
                crc^=0xEDB88320;
            }
        }
        crc_32_tab[val] = crc;
    }
}

// This is based on the CRC routine used in the ATOC atoccrc
// program.  This procedure was also used in calculating the
// original HIFT file CRC values and so the basic preocedure
```

```
// is retained in this program.
//
unsigned long CalcCRC(char* cFileName)
{
    unsigned long crc, byte, mask=255L;
    FILE *input;

    if ((input = fopen(cFileName, "rb")) == NULL) {
        fprintf(stderr, "can't open input file %s\n", cFileName);
        exit(1);
    }

    // Start CRC value not 0 to allow differing length null character
    // files to generate different CRC values.
    //
    crc=~0L;
    if (cCRCMode == 'h') crc = 0; // But didn't originally do this.

    while (!feof(input)) {
        byte=(unsigned long)fgetc(input);
        if (feof(input)) continue;
        crc = crc_32_tab[(crc^byte) & mask] ^ (crc >> 8);
    }
    fclose(input);

    return crc;
}

// Check CRC values given on a list.
// Normal files have CRC entries only with leading space.
// Original HIFT CRC files are supported.
//
void CheckCRCs(char* CRCFile)
{
    char cFileName[BufferSize+1];
    char cFullName[BufferSize+1];
    unsigned long inCRC, nowCRC;
    FILE *input;

    if ((input = fopen(CRCFile, "ra")) == NULL) {
        printf("can't open CRC file %s\n", CRCFile);
        exit(1);
    }

    while (feof(input)==0) {

        if (fgets(cTemp, BufferSize, input) == NULL) continue;

        // skip lines not starting with a space
        //
        if ((*cTemp != ' ')&&(*cMode != 'h')) continue;
```

```
        if (*cMode == 'h') {
            sscanf(cTemp, "%s %lx", cFileName, &inCRC);
        }
        else {
            sscanf(cTemp, "%lx %s", &inCRC, cFileName);
        }
        cpT = cFileName;
        while (*cpT != '\0') {
#ifndef WIN32
            if (*cpT == '\\') *cpT = '/';
#endif
            if (cCase == 'l') *cpT = tolower(*cpT);
            if (cCase == 'u') *cpT = toupper(*cpT);
            cpT++;
        }

        lFileCount++;
        strcpy(cFullName, cBasePath);
        if (*cMode == 'h') {
            strcat(cFullName, cFileName);     // no prefix for early hift
        }
        else {
            strcat(cFullName, &cFileName[2]); // skip the ./ prefix
        }
        nowCRC = CalcCRC(cFullName);
        if (nowCRC != inCRC) {
            printf("was: %08lX  now: %08lX  %s\n", inCRC, nowCRC, cFileName);
        }
    }

    fclose(input);
}
```