

# Durham E-Theses

---

## *Temporal variation in the radio environment*

Roger Lewenz

### How to cite:

---

Lewenz, Roger (2005) Temporal variation in the radio environment. Doctoral thesis, Durham University.

### Use policy

---

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a <https://etheses.durham.ac.uk/id/eprint/2726/> is made to the metadata record in Durham E-Theses
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full Durham E-Theses policy](#) for further details.

## **APPENDIX 5. CONTENTS OF CDROM**

### **A5.1 INTRODUCTION**

This appendix lists the contents of the CDROM included with this thesis. The programs which have been used for processing the measurement data are included with a brief description of the purpose and method of building and using each of them. For each program the source code, executable and ancillary files have been written on the CDROM. All of the programs have been written in the C programming language using the Kernigan and Ritchie dialect rather than the ANSI dialect. This ensures that simple compilers may be used and the programs may be built and run on a variety of computers. The programs have been run on 486 and Pentium class PCs and on Hewlett Packard 9000 series machines. The programs have been built using the Zortech V3 compiler with 32 bit extender (-mx compiler switch), the GNU C compiler and the Hewlett Packard compiler provided as part of the HP-UX 10.20 operating system. On a PC the programs may be built and run under MS-DOS and in a command window under Windows NT 4.0. The programs are unlikely to run under Windows 2000 since this operating system appears to have an aversion to running programs not developed by Microsoft.

There are Delphi versions of some of the programs. These have been developed with Borland Delphi 5 and provide a Windows interface. These programs will run under Windows 95, Windows 98 and Windows NT 4.0.

The programs used with the Antenna Characterisation rig were initially written in C to run on 16 bit systems. They have been translated into Delphi to provide a Windows interface but because the programs address the hardware directly they were built with Borland Delphi 1 (16 bit).

### **A5.2 pk2pk4**

When using the sounder with an array of directional antennas it was necessary to determine the relative gains of the eight channels. The method of doing this is described in Chapter 6 paragraph 6.2 and the result is a 10 sweep measurement

file. The program **pk2pk4** converts the 10 sweep measurement file into a file containing the relative gains.

If the measurement yield the file `example.dac` then by convention the 10 sweep file from relative gain measurement is called `examplex.dac`. This is converted into a file called `example.ini` using **pk2pk4**. `example.ini` is read by the channel splitting program **splitchi**.

The following command line is used to effect the conversion

```
pk2pk4 examplex.dac example.ini
```

The files for the C version are:

```
\pk2pk4\C\pk2pk4.c  
\pk2pk4\C\posmean.c  
\pk2pk4\C\negmean.c  
\pk2pk4\C\pk2pk4.exe
```

The file for the Delphi version are:

```
\pk2pk4\delphi\pk2pk.cfg  
\pk2pk4\delphi\pk2pk.dof  
\pk2pk4\delphi\pk2pk.dpr  
\pk2pk4\delphi\pk2pk.exe  
\pk2pk4\delphi\pk2pk.res  
\pk2pk4\delphi\unit1.dcu  
\pk2pk4\delphi\unit1.dfm  
\pk2pk4\delphi\unit1.pas
```

### **A5.3 splitchi**

The **splitchi** program is used to split a measurement file from the sounder into individual channel files and split the upper and lower bands if both band were used. This program silently reads a file with relative gains output by `pk2pk4` if it exists in the same directory as the input file. The following command line was used to invoke the program:

```
splitchi input_filename output_file_basename s_flag
```

where

`input_filename` is the name of the measurement file to be split e.g.  
`example.dac`.

`output_file_basename` is the part of the names of the output files before the dot.  
The program will add filename extensions `l1, l2, l3, l4, l5, l6, l7, l8` and `u1, u2, u3, u4, u5, u6, u7, u8` for the single channel files containing the lower band and upper band data.

`s_flag` is a single letter indicating whether band splitting is required and which band is first in the input file. 'L' indicates that the bands are to be split with the lower band occurring first in the input file. 'U' indicates that band splitting is required and the upper band data is first in the input file. 'N' indicates that band splitting is not required.

The files for the C version are:  
`\splitchi\C\splitchi.c`  
`\splitchi\C\splitchi.exe`

The files for the Delphi version are:  
`\splitchi\delphi\splitchn.dpr`  
`\splitchi\delphi\splitchn.exe`  
`\splitchi\delphi\splitchn.opt`  
`\splitchi\delphi\splitchn.res`  
`\splitchi\delphi\unit1.dcu`  
`\splitchi\delphi\unit1.dfm`  
`\splitchi\delphi\unit1.pas`

#### **A5.4 delay**

The **delay** program reads a single channel file from the **splitchi** program and writes a file containing a power delay profile averaged over all the sweeps in the input file. There are three variants, **delay** has an independent axis scaled for delay, **fdelay** has an independent axis scaled for beat frequency and **bdelay** has an independent scales for bin number. All three programs produce an output file which is a text file suitable for reading and plotting with Microsoft Excel. These programs use the FFTW Fourier transform routines from MIT which will be found in the `\fftw` directory. Any of the programs may be invoked with two command line arguments thus:

```
delay input_file output_file
```

The files are:

```
\delay\C\delay.c  
\delay\C\delay.exe  
\delay\C\fdelay.c  
\delay\C\fdelay.exe  
\delay\C\bdelay.c  
\delay\C\bdelay.exe.
```

### **A5.5 FFTW**

This is the FFTW Fourier transform package from MIT. The `\fftw` directory contains the source code and makefiles in `\fftw\src`, the header file in `\fftw\include`, documentation in `\fftw\doc` and compiled library files in `\fftw\lib`. Two library files are included: `fftw.lib` for use with the Zortech compiler and `libfftw.a` for use with the GNU compiler, both for machines with Intel X86 processors.

### **A5.6 delrun2**

This program reads a single channel file from `splitchi` and writes a file containing the mean delay and RMS delay spread for each 5 sweeps in the input file. The program allows a threshold to be set and only components with more than the threshold power are considered. The output file is a text file which may be read by Microsoft Excel. The program is invoked with the command line:

```
delrun2 input_file output_file threshold
```

The threshold is expressed in dB and uses the power in the strongest component as datum.

There is only a C version of this program. The files are:

```
delrun2.c  
delspred.c  
makefile
```

This program needs FFTW.

### **A5.7 pdfcdf**

This directory contains four programs which read the output of **delrun2**.

**meanpdf** produces a PDF of the mean delays in a file which may be plotted with Microsoft Excel.

**meancdf** produces a CDF of the mean delays in a file which may be plotted with Microsoft Excel.

**rmspdf** produces a PDF of the RMS delay spread in a file which may be plotted with Microsoft Excel.

**rmscdf** produces a CDF of the RMS delay spread in a file which may be plotted with Microsoft Excel.

The programs are called from a command line with two arguments, the first is the name of the input file, the second is the name of the output file.

e.g.

```
meanpdf input_file Output_file
```

All of the programs need the subroutine contained in `meanset.c` and expect in initialisation file `meanset.ini` to exist in the current working directory. An example of `meanset.ini` is included in the directory. This file specifies the start point and the increment of the delay axis.

The files are:

```
meanpdf.c  
meancdf.c  
rmspdf.c  
rmscdf.c  
meanset.c.
```

## **A5.8 findcom4**

This program reads a file of time series data for a single channel output by the **splitchi** program and writes a file with:

Time from start of measurement run

Number of multipath components

Minimum delay

Maximum delay

for each 5 sweeps of the input data. The output file is a text file which may be examined with Microsoft Excel or Notepad. This file is usually input to the **compcdf** program which produces a CDF of the number of components. The

program has four command line arguments: the name of the input file, the name of the output file, the threshold to be set, the leakage value to be set.

The files are:

`findcom4.c`

`findcmp.c`

`makefile`

This program needs FFTW.

### **A5.9 compcdf**

This program reads the output from the `findcom4` program and produces a CDF of the number of components. The median and mean number of components are also output on the standard output. The program has two command line arguments: the name of the input file, the name of the output file.

The files are:

`compcdf.c`

`bubble.c`

### **A5.10 kolmogor**

This program reads two files output from `delrun2` and calculates the Kolmogorov-Smirnov difference and probability statistics. The directory contains all the files needed to build a Delphi version of the program.

### **A5.11 gaussian**

This directory contains three programs: `optmean`, `optmean2` and `optsigma`. These programs search for the best mean and standard deviation parameters of a Normal distribution to fit the measurement data. The input file is the output from `delrun2`. The `optmean2` program outputs a file containing the CDF of the Normal distribution which best fits the data from the measurement.

### **A5.12 weibull**

This directory contains three programs: `optshape`, `optshape2` and `optscale`. These programs search for the shape and scale parameters of a Weibull distribution which best fits the measurement data. The `optshape2` program outputs a file containing a CDF of the Weibull distribution which best fits the measurement data. All the programs read a file output from the `delrun2` program.

### **A5.13 pdelay3**

The **tseries** directory contains all the files needed to build the **pdelay3** program. This is a Delphi program which may be used to examine a single channel time series file from the **splitchi** program, 5 sweeps at a time. This program was used to detect faults in the recorded data such as occurred during the measurement on 4 April 2003 in Whitworth Street in Manchester.

### **A5.14 newdac10**

The **newdac10** directory contains all the files needed to build the new simple data acquisition program **sounder1**. This is a Delphi program which acquires 3 channels of data using a single PCI-DAS4020/12 data acquisition card.

### **A5.15 antcalib**

The **antcalib** directory contains files from which may be built a number of programs for antenna characterisation. These programs control the antenna characterisation rig and record the output of the various receivers. Some of the programs were written in C with assembler subroutines to run in a 16 bit environment. They were build with the Zortech version 1 C compiler and the Microsoft assembler MASM version 4. These programs will run under MS-DOS or in a DOS box under Windows 95. The remainder of the programs were written in Delphi to run in a 16 bit environment. They were build with Borland Delphi version 1 and should run under Windows 3.1 or Windows 95. They will not run under Windows NT. These programs should be described in M. Abdalla's thesis[43].