

## APPENDIX D

# EXAMPLE DST ANALYSIS PROGRAM

The following C source code provides an example program for the analysis of AVHT and/or GEANT DST files. With it you can analyze them independently or comparatively. The given example creates the probability and  $\chi^2_\nu$  distributions upon which Figs. 3-12 and 4-1 are based. The program can be easily modified to create many other distributions, as described in *stat.h*.

### D.1 Dstanal.c

```

/*****
DSTAnal.c

DST Analysis Program
Analyzes AVHT DSTs, GEANT DSTs or both at once.

By Erik Kangas
Last Updated 4/29/95

*/

#include "hougher.h"          /* AVHT Algorithms */
#include "stat.h"             /* Statistics functions */
#include <stdlib.h>

/*****
Specialized tracking variables
You can put your own global variables here

det_name  -> name of the detector data file
dst_name  -> name of the AVHT dst data file (ignore if not used)
gdat_name -> name of the GEANT dst data file (ignore if not used)
out_file  -> base name for the output analysis files.
dstf      -> AVHT DST file
gdstf     -> GEANT DST file

*/

#ifdef THEORY3
char det_name[80] = "/minimax/kangas/tracking/data/new24.dat";
char dst_name[80] = "/minimax/kangas/tracking/data/0327nall.dst";
char gdst_name[80] = "/minimax/kangas/tracking/data/0327.gdt";

#elif defined(DOS)
char det_name[80] = "E:\\minimax\\data\\new24.dat";
char dst_name[80] = "E:\\minimax\\data\\0327nall.dst";
char gdst_name[80] = "E:\\minimax\\data\\0327.gdt";

```

```

#endif

char out_name[80] = "0327";
FILE *dstf, *gdstf;

/*****
Definitions of which DST files to use in analysis

if NOHDST is defined, the AVHT DST will not be used in the analysis
if NOGDST is defined, the GEANT DST will not be used in the analysis
If both are NOT defined, both will be used and the events will be synced
so that the same event from both is being analyzed at the same time.

In this case, both are undefined, so both will be used.
*/

/*
#define NOGDST

#define NOHDST
*/

void Analysis(void) {
    int evtnum=0,                /* Count number of AVHT events processed */
        evts=0;                 /* Count number of GEANT events processed */
    Tracks *gcut=NULL,          /* Storage for GEANT event information */
           *gh=NULL,
           *gt=NULL,
           *htracks=NULL,      /* Storage for AVHT event information */
           *th=NULL,
           *tt=NULL;
    Groups *ggrp=NULL,         /* Geant Track Groups */
           *tgrp=NULL;        /* AVHT track groups */

#ifdef NOGDST
    gcut = (Tracks*) malloc(sizeof(Tracks));
    gh = (Tracks*) malloc(sizeof(Tracks));
    gt = (Tracks*) malloc(sizeof(Tracks));
    ggrp = (Groups*) malloc(sizeof(Groups));
#endif

#ifdef NOHDST
    htracks = (Tracks*) malloc(sizeof(Tracks));
    th = (Tracks*) malloc(sizeof(Tracks));
    tt = (Tracks*) malloc(sizeof(Tracks));
    tgrp = (Groups*) malloc(sizeof(Groups));
#endif

    if defined(NOHDST) && defined(NOGDST)
        return;                /* Nothing to do */
    endif

    /* Use SetupStatistics to define the types of automatic data analysis
    /* to be used. See stat.h for descriptions of the possibilities.
    /* To do other types of analysis, you should define you histograms and
    /* other variables here.
    /* In this case, we will make distributions of the probability of
    /* heads coming from C0 and distributions of the chi^2_v of tracks
    /* found and tracks spurious.

    SetupStatistics(stChi);

#ifdef NOHDST

    /* This handles loading events from only a GEANT DST file */

```

```

/* when an AVHT DST is not going to be analyzed */

while ( LoadGeantDST(gcut,gdstf) == 0) {
    SliceGeant(gcut,gh,gt,ggrp);      /* Get heads & tails in the event */
    evts++;                            /* Count the events */
    if (!(++evtnum)%10)
        printf("Event %d, %d\n",evtnum,gcut->event);
}

#elif defined(NOGDST)

/* This handles loading events from only an AVHT DST file */
/* when the GEANT DST is not to be used */

while ( LoadTEvent(dstf,htracks) == 0) {
    SplitTracks(htracks,th,tt);      /* Get separate lists of heads&tails */
    GroupTracks(th,tt,tgrp);        /* Group and cut AVHT tracks */
    if (!(++evtnum)%10)
        printf("Event %d, %d\n",evtnum,htracks->event);
}

#else

/* This loads events from both AVHT and GEANT DST files and makes sure */
/* the current events being analyzed are the same. It assumes that */
/* The events in the AVHT DST are a subset entirely contained in the */
/* GEANT DST. */

while ( LoadTEvent(dstf,htracks) == 0) {
    if (!(++evtnum)%10)
        printf("Event %d, %d\n",evtnum,htracks->event);

    do {
        if (LoadGeantDST(gcut,gdstf) < 0) return;
        evts++;
    } while (htracks->event != gcut->event);

    SliceGeant(gcut,gh,gt,ggrp);      /* Get heads and tails in GEANT DST */
    SplitTracks(htracks,th,tt);      /* Get heads and tails from AVHT DST */

    GroupTracks(th,tt,tgrp);        /* Group and cut AVHT tracks */
}

#endif

/* Now we have the AVHT and / or GEANT tracks for the current event */
/* We may perform any type of analysis we choose. */
/* In this case, we are only doing automatic data analysis, so we */
/* call DoStatistics with the tracks. Note: is we are not using one*/
/* type of DST file, we should pass NULL for the relevant tracks and */
/* be sure that we are not doing analysis that requires them. */

DoStatistics(th,tt,gh,gt,ggrp);
} /* end event loop */

/* All the events have been processed */
/* Output the number of AVHT and GEANT events there were */

printf("Nevents :%d AVHT, %d GEANT\n",evtnum,evts);
}

int main() {
    PRECOMPUTE=0;                    /* Do not allocate memory for tracking */
    setup_hougher_(det_name);        /* Setup the tracking environment */
    OutFileBase(out_name);          /* Setup the base output file name */

#ifdef NOHDST                        /* Open the AVHT DST file */
    if (!(dstf = fopen(dst_name,"r"))) {

```

```

        printf("AVHT DST file '%s' not found!\n",dst_name);
        exit(1);
    }
#endif

#ifdef NOGDST /* Open the GEANT DST file */
    if (!(gdstf = fopen(gdst_name,"r"))) {
        printf("GEANT DST file '%s' not found!\n",gdst_name);
        exit(1);
    }
#endif

    Analysis(); /* Analyze the events */
    return 1;
}

```

## D.2 Makefile

```

#
# AVHT Makefile for use with non-OFFLINE analysis
#
# By Erik Kangas -- April 29, 1995
#
#
# Insert the base name of your 'C' interface file below
# This is the only section you should have to modify.
# You may alter the CFLAGS variable
#

CFILE = dstanal

#
# Following are definitions necessary for compilation / linkage
# - The path to the source files
# - Compiler directives
# - Malloctype corrects for errors in Theory3's compiler
#

HOUGHDIR = /minimax/kangas/tracking/source/

MALLOCTYPE=3.1; export MALLOCTYPE

CCOMP = xlc
CFLAGS = -O -I$(HOUGHDIR) -DTHEORY3

.c.o:
    $(CCOMP) -c -o $*.o $(CFLAGS) $<

#
# Define our object files
#

OBJECTS = $(CFILE).o $(HOUGHDIR)bjhough.o $(HOUGHDIR)nrc.o\
    $(HOUGHDIR)trackdef.o $(HOUGHDIR)hougher.o $(HOUGHDIR)coords.o\
    $(HOUGHDIR)stat.o

#
# Compile / link the files. Use fortran linker.
#

$(CFILE) : $(OBJECTS)
    $(CCOMP) $(CFLAGS) -o $(CFILE) $(OBJECTS) -lm

```

```
#
# File dependencies
#
ALLDEP = $(HOUGHDIR)hougher.h $(HOUGHDIR)trackdef.h $(HOUGHDIR)bjhough.h\
        $(HOUGHDIR)coords.h $(HOUGHDIR)stat.h $(HOUGHDIR)nrc.h

$(CFILE).o: $(CFILE).c $(ALLDEP)

$(HOUGHDIR)bjhough.o: $(HOUGHDIR)bjhough.c $(ALLDEP)

$(HOUGHDIR)hougher.o: $(HOUGHDIR)hougher.c $(ALLDEP)

$(HOUGHDIR)stat.o: $(HOUGHDIR)stat.c $(ALLDEP)

$(HOUGHDIR)coords.o: $(HOUGHDIR)coords.c $(HOUGHDIR)coords.h $(HOUGHDIR)nrc.h

$(HOUGHDIR)nrc.o: $(HOUGHDIR)nrc.c $(HOUGHDIR)nrc.h

$(HOUGHDIR)trackdef.o: $(HOUGHDIR)trackdef.c $(ALLDEP)
```