CB–Note 143

LEAR Crystal Barrel Experiment, PS197
$r\phi$– and dE/dx–Calibration of the JDC
JDC Calibration Version 1.02/00

Curtis A. Meyer
Universität Zürich

15 November, 1990

Contents

1 Introduction

This note is a technical description on how the $r\phi$–calibration of the JDC is performed. It is assumed throughout this manual that the user is familiar with the Crystal Barrel Reconstruction Software [1] and in particular the Chamber Reconstruction Software [2], [3]. It is also assumed that the user is familiar with the MINUIT program [5] as well as HBOOK [6] and HPLOT [7] Finally, it is assumed that the user is familar with CMZ [9]. This manual is not intended for the general users; rather it is meant as a detailed guide for people wishing to perform an $r\phi$–calibration of the JDC. The style of this manual follows that of the Z–calibration manual [4].

Before starting the calibration, it is necessary that the user have the correct version of LOCATER. The code should be generated using the *CALIB cradle as described in Building the LOCATER code in reference [2].

There are several different programs used to perform the $r\phi$–calibration of the JDC. The first is a set of USER subroutines which are linked with the Crystal Barrel Offline software and makes a program known as CBCIRC. CBCIRC generates several output files which are used as input to an external code, FITRPHI. In addition to this USER code, there is a plotting package which examines the resolution seen in the chamber at every iteration, PLOT_CIRC. Finally, there is the external code, FITRPHI, which performs a MINUIT fit to the output data from CBCIRC, and creates the next iteration of the calibration. This document describes how to create all of these programs, and how they are then run to generate a calibration for the JDC.

2 The $r\phi$–Calibration Software

2.1 User Code for $r\phi$–Calibration

The USER code for $r\phi$–calibration is distributed as a CMZ card file, USER_CALB.CAR. This should be made into a CMZ file, USER_CALB.CMZ by the user. (Note, the CMZ file contains all JDC calibration programs. In principle, all needed card files, kumac files, and command files are stored in the CMZ file, however much of that information will be repeated here.

In order to perform the $r\phi$–calibration, the user will need to create several programs from routines stored in the CMZ file. The main program is known as CBCIRC. This program tracks the data through the circle fit stage of LOCATER, and then drops all tracks which do not cross sector boundaries. The remaining tracks are then refit with the circle fit, but this time the coordinates, r and ϕ are allowed to change. The fit r and ϕ are then converted into x and y coordinates, and this data as well as the wire number and drift time are written to a large external file for later input to a MINUIT fit.

In order to create and run this program, the user will need the following KUMAC files, and the following command procedures. These should be extracted from the CMZ file for the user's machine by doing the following under CMZ.

```
CMZ> file user_calb
user_calb> cd shell_script
user_calb> select machine_flag (ALT,DECS,IBM,SUN,VAX)
user_calb> set user_calb.kumac -D
user_calb> ctot -Y kuma_cod
user_calb> set plot_calb.kumac -D
user_calb> ctot -Y kuma_plt
user_calb> set calb.exec -D
user_calb> ctot -Y exec_cod
user_calb> set plot.exec -D
user_calb> ctot -Y exec_plt
```

```
user_calb> exit
```

This will create the files described below. The provided listings are specific to the Alliant FX/8 and are only given as examples.

- USER_CALB.KUMAC which is a generic KUMAC file for creating any CBCALB calibration user program.

```
file carlink/cboff.cmz -R
file user_calb.cmz     -R
sel .
seq .
pilot .
select ALT        (Specify the desired machine flag.)
select SEP_90_1   (Specify the particular calibration set.)
seq   //cboff/commcb
cd    //user_calb
pilot //user_calb/*circcod
set src/*.f -F
ctof -P
release *
```

- PLOT_CALB.KUMAC which is a generic KUMAC file for creating any PLOT_CALB plotting program.

```
file user_calb.cmz -R
set plot_calb.f -F
select .
select ALT    (Specify your machine flag here.)
pilot  .
pilot  *cirplt0
ctof -P
release *
```

- CALB.EXEC which is a generic command procedure for running any CBCALB program. Consult the next section for an example of this file.
- PLOT.EXEC which is a command procedure for running either and PLOT_CALB program. Consult the next section for an example of this file.

In order to create the CBCIRC program, change the pilot in the in the USER_CALB.KUMAC file to *CIRCCOD. To create a plotting program, change the pilot in the PLOT_CALB.KUMAC file to be one on the following:

- *CIRPLT0 This is a general set of pictures. Initially, there is a picture of the χ^2 and the probability of each fit track. Then plots showing general track data. Finally, there are 23 pictures, one for each layer in the JDC. Each of these has eight plots on it, four from the left side of the drift cell, and four from the right side. The four plots are: [1] the value of $r \cdot d\phi$, [2] the fit error in $r \cdot d\phi$, [3] the pull in $r \cdot d\phi$, and [4] the values of ϕ used.
- *CIRPLT1 This is a set of 23 pictures contains plots showing the resolution acheived. It has four plots per picture, two from the left and two from the right side of the cell. These two pictures are: [1] $r \cdot d\phi$ versus ϕ and, [2] $r \cdot d\phi$.

- *CIRPLT2

The programs are now created by excuting the KUMAC files from within CMZ

```
CMZ> exec user_calb.kumac
CMZ> exit
```

2.2 The External Calibration Code

In addition to the programs described in the previous section, there is a second file which is used for $r\phi$–calibration. This is available as a CMZ card file, FITRPHI.CAR, which should be converted to a CMZ file, FITRPHI.CMZ. This file is used to generate an external fitting program, FITRPHI. First one needs to extract the correct KUMAC file from the CMZ file.

```
CMZ> file fitrphi.cmz
fitrphi> select machine_flag, (ALT, DECS, IBM, SUN, VAX).
fitrphi> cd shell_script
fitrphi> set fitrphi.kumac -D
fitrphi> ctot -Y kumac
fitrphi> set fitrphi.exec -D
fitrphi> ctot -Y exec_rphi
fitrphi> exit
```

Make sure that the pilot selected in the FITRPHI.KUMAC file is *FITREAL, and then do the following:

```
CMZ> exec fitrphi.kumac
CMZ> exit
```

The resulting code is linked with the KERNLIB, PACKLIB, and GRAFLIB parts of the CERN library.

- FITRPHI.KUMAC is the KUMAC file used to create this program. The Alliant FX/8 version is shown here.

```
file fitrphi -R
*
select .
select ALT
select BPLUS   (or BMINUS, choose direction of magnetic field).
*
pilot .
pilot *FITREAL
*
set srcfitrphi/*.f -F
*
ctof -P
*
release fitrphi
```

- FITRPHI.EXEC is the command procedure used to run the external calibration. An example of this is given in the next section.

3 Running of the Calibration Code

The process of calibration is an iterative procedure which consists of running through the previous programs several times. In order to perform a calibration, a sample of minimum bias data consisting of 30000 to 40000 events is needed.

This section will describe how to run each of the programs needed for calibration. It will not describe how the calibration is performed; that will be described in the next section. What is detailed is the input for each calibration program, the important output from each program, and a shell script for running each program. All shell scripts are written in c–shell; the user will need to obtain the scripts for other machines as detailed in the previous sections.

3.1 CBCIRC

In order to run CBCIRC as a calibration program, (and not as a general reconstruction program), one needs to tell the CBOFF code what is going on. This is done by using the following card file for CBOFF, (see Data Cards in reference [1]). The user should be very careful with USER card 7. During the first iteration, this needs to be set to $+1$ if the magnetic field is along the $+z$ axis, and to -1 if the magnetic field is along the $-z$ axis. If these get mixed up, you will be unable to perform a calibration. For later iterations, this card absolutely must be set to 0. Before using the following, be sure to strip off the comments.

```
WRIT 4         ! Echo this file to logical unit 4.
LIST           ! Echo this file.
FZIN  'XT'  0 ! Read data in from tape.
ZFOUT 'XTO' 0 ! Write dst to tape.

USER 6=40000  ! Analyze at most 40000 events.
USER 7=-1     ! Select field direction negative - default table.
XTAL 'NONE'   ! Turn off crystal reconstruction.
GLOB 'NONE'   ! Turn off global tracking.
STOP  <===== READING STOPS HERE ========
    USER(1)   Number of events to analyze.
    USER(2)   First event to analyze.
    USER(3)   Event to turn on Debug.
    USER(4)   First run to analyze.
    USER(5)   Last run to analyze.
    USER(6)   Total number of events to use.
    USER(7)   Set to zero for normal table.
              Set to +1 for starting table for positive field.
              Set to -1 for starting table for negative field.
```

Next, one has to direct LOCATER to perform a calibration. This is done using the following card file, which is assigned to logical unit 81. This card file is also valid only for the first iteration. Please note that you need to strip off the comments before using.

```
WRIT 4                ! Echo to unit 4.
LIST                  ! Echo!

DELZ  23*0.90         ! Z-resolution.
ANGL    90.00  1.0    ! Angle offset to sector 1.
IAMP   400     1      ! JDC pulse height cut.
```

```
BMAG   -15.    1.0     ! Magnetic field strength.

LGT2   T               !

JDCL   T               ! Do JDC calibration.
ICAL   0               ! Calibration type 0.
LGWR   1               ! Bad wires in z are not used.

MCIR   12              ! Minimum hits per track.
ZOFF   0.00  1.00      ! Offset to target center = 0.0
ZPRB   0.000           ! Probability cut to keep a track.
CLIM  40.0             ! Chisquare limit not to drop points
CFRA  0.25             ! Drop a point if it is 25% of chisquare.
CSMX  0.2618           ! Cos(alpha_max) for isochron corrections.
DYMX  0.2500           ! The maximum a point can shift and be written

SY02   2.00E-4         ! Constant term in resolution function.
SYDF   2.00E-4         ! Term proportional to sqrt(distance).
SYDQ   0.0289          ! Special linear error term.
SXDQ   0.0524          ! Special error term in radius.

STOP  <======= READING STOPS HERE ====
    DEBG(1) : Overall debug control
    DEBG(2) : Debug unpacking raw data.
    DEBG(3) : Debug pattern recognition.
    DEBG(4) : Debug circle fits.
    DEBG(5) : Debug helix fits.
    DEBG(6) : Debug vertex fits.
    DEBG(10): Debug the slow control processing.
```

Both of these card files can be obtained from the CMZ file by the following:

```
CMZ> file user_calb.cmz
user_calb> cd circ_instruct
user_calb> set card_circ.crd -D
user_calb> ctot card_crd
user_calb> set card_circ.jdc -D
user_calb> ctot card_jdc
user_calb> exit
```

The following c–shell script is used to run the CBCIRC program on the Alliant FX/8 in Zürich. Versions for other machines can be obtained as described previously.

```
#       Filename:      cbcalb_run
#       Language:      Berkeley Unix, C-shell
#       Author:        Curtis A. Meyer
#       Creation date: 10 February, 1990
#       References:
#       Description:   This file will run the cbzfit code.
#       Input:   card_circ.crd   Steer CBOFF
#                card_circ.jdc   Steer LOCATER
```

```
#               jdc_gain.tbl    JDC Gain Table.
#               FOR021          Raw Data Tape
#       Output:  circ.log        Status of the run.
#               circ.dbg        Possible debug output.
#               circ.err        Errors encountered in running.
#               circ.hist       Histogram output, (input to plot_zfit).
#               CALCIRCLE.DAT    Calibration data written by the program,
#                               (input to fitrphi).
#               JDCSLCREF.DAT   Calibration data written by the program
#                               containinf slow control information,
#                               (input to fitrphi).
#     ##########################################################
#
#---Assign the output files.---
   setenv FOR004 circ.log
   setenv FOR007 circ.dbg
   setenv FOR008 circ.err
   setenv FOR010 circ.hist
#---Assign the input card files, (cf FFREAD).---
   setenv FOR005 card_circ.crd
   setenv FOR081 card_circ.jdc
#---Assign the jdc_gain file.---
   setenv FOR082 jdc_gain.tbl
#---Assign the raw data tape.---
   setenv FOR021 /dev/exabyte0
#---Run the program---
   cbcirc
#---Deassign all units:---
   unsetenv FOR003
   unsetenv FOR004
   unsetenv FOR007
   unsetenv FOR008
   unsetenv FOR010
   unsetenv FOR011
   unsetenv FOR020
   unsetenv FOR080
   unsetenv FOR081
   unsetenv FOR082
   exit(0)
```

There are four important output files from CBCIRC. The first are the files CALCIRCLE.DAT and JDCSLCREF.DAT which are used as input to the FITRPHI program. These contain packed data containing the wire number, drift times, and fit x,y coordinates for all accepted hits, and averaged slow control information over the calibration period. The next file is CIRC.LOG, the log file from this program. This file gives information on how well everything was fit, how many tracks were found in each sector, and how often each wire was used in the calibration data. Finally, there is a file CIRC.HIST, which is the output from HBOOK. This file is read in and examined by the PLOT_CIRC program.

3.2 PLOT_CIRC

The PLOT_CIRC program reads in the CIRC.HIST file as generated by CBCIRC, and plots the reults. These can then be printed and used to monitor the progress of the calibration. The only useful output from this program is either METAFILE.OUT or POSTSCRIPT.OUT, (depending on your machine). This file should be printed on the appropriate graphics device. The following shell script is used on the Aliant FX/8 to run this program.

```
#
#      Assign logical unit 003 to the zfit.hist file.
#
    setenv FOR003 /users/cmeyer/CB/cboff/circ.hist
#
#      Run the plotting program.
#
    plot_circ
#
#      Deassign the unit
#
    unsetenv FOR003
#
    exit(0)
```

3.3 FITRPHI

The FITRPHI program reads in the CALCIRCLE.DAT and JDCSLCREF.DAT files written by CBCIRC, and performs a MINUIT fit on every layer to the drift–time parametrizations. The output from this program consists of a FORTRAN subroutine, and sevral logfiles. Note, it is absolutely critical that the correct magnetic field direction was chosen when this code was created.

- NEW_TABLE.F is a fortran callable subroutine which needs to be linked with the next iteration of the CBCIRC code.
- FIT.CHI is a file telling you how well each side of each layer was fit.
- FIT.OUT contains a list of the fit parameters, and the χ^2 for each fit performed.
- FIT.LOG contains all the messages printed by MINUIT during each fit (HBOOK performs many Gaussian fits in this package. For these fits, HBOOK uses MINUIT.)

The following shell script is used on the Alliant FX/8 to run the FITRPHI program. Versions for other machines can be extracted as described previously.

```
#
#     File:          fit_rphi.sh
#     Author:        Curtis A Meyer
#     Creation date: 4 July,1989
#
#     ########################################
#
   set Home=$cwd
#
#     Set up the input files:
```

```
#
   setenv FOR034   cbdata/CALCIRCLE.DAT
   setenv FOR035   cbdata/JDCSLCREF.DAT
#
#     Set up the output files:
#
   setenv FOR003  fit.log
   setenv FOR004  fit.out
   setenv FOR013  fit.dbg
   setenv FOR015  table.f
   setenv FOR017  fit.chi
#
#     Run the program
#
   fitrphi
#
   unsetenv FOR003
   unsetenv FOR013
   unsetenv FOR015
   unsetenv FOR017
   unsetenv FOR034
#
   sed 's/^ //g' table.f > new_table.f
   rm -f table.f
#
   exit(0)
```

4 $r\phi$–Calibration of the JDC

This section describes the steps involved in performing an $r\phi$–calibration of the JDC. The section is divided into units called iterations, where each iteration will involve running some or all of the previous code.

4.1 Iteration 1

In this iteration, one starts with the default calibration table as selected via USER card 7. For a postive magnetic field this card must be +1 and for a negative magnetic field, this card must be −1. During this iteration, one also writes a PDT, (Processed Data Tape), which is then used as input to later iterations. The PDT contains *processed* rather than *unprocessed* JDC data. This speeds up analysis of later iterations by about a factor of three.

Step 1:
Run the CBCIRC program using the input cards CARD_CIRC.CRD and CARD_CIRC.JDC as described earlier. You will need to process about 40000 events in order to do a good calibration.

Step 2:
Run the FITRPHI code using the output files CALCIRCLE.DAT and JDCSLCREF.DAT from step 1 as input.

Step 3:
Print out the FIT.CHI file. Then edit the FIT_TABLE.F program, and change the line.

```
+DECK,MKREAL.
```

to

```
+DECK,MKmmyy_i,IF=MMM_yy_i.
```

Where *mm* is the number of the month in which these data were taken, (04 for April), *yy* is the year in which these data were taken, (90 for 1990), *i* is the calibration set number for this month and year, (normally 1, unless conditions changed and an additional calibration is needed, and *MMM* is the name of the month, (APR for April). As an example:

```
+DECK,MK0490_1,IF=APR_90_1.
```

Step 4:
Now include the modified FIT_TABLE.F code in the USER_CALB.CMZ file. This is done by doing the following:

```
CMZ [0] file user_calb
user_calb [1] cd circcode
user_calb [2] ytoc fit_table.f
user_calb [3] exit
```

Now you need to create a new version of CBCIRC selecting the DECK flag that was created in this step. An example of FIT.CHI after the first iteration is the following:

```
 Lyr/side  1 1 Chsq  10.78399    5883 sigma  0.042814
 Lyr/side  1 2 Chsq   9.63814    5106 sigma  0.043447
 Lyr/side  2 1 Chsq  10.72137    7100 sigma  0.038859
 Lyr/side  2 2 Chsq   9.52930    6006 sigma  0.039833
 Lyr/side  3 1 Chsq  15.46660    7766 sigma  0.044627
 Lyr/side  3 2 Chsq  12.47319    6913 sigma  0.042477
 Lyr/side  4 1 Chsq  20.92821    7266 sigma  0.053668
 Lyr/side  4 2 Chsq  19.03415    6955 sigma  0.052314
 Lyr/side  5 1 Chsq  26.18241    7675 sigma  0.058407
 Lyr/side  5 2 Chsq  21.81757    6833 sigma  0.056506
 Lyr/side  6 1 Chsq  27.26429    7714 sigma  0.059451
 Lyr/side  6 2 Chsq  27.43186    7990 sigma  0.058594
 Lyr/side  7 1 Chsq  27.44385    8052 sigma  0.058381
 Lyr/side  7 2 Chsq  24.47470    7656 sigma  0.056540
 Lyr/side  8 1 Chsq  22.55975    7488 sigma  0.054889
 Lyr/side  8 2 Chsq  23.28872    8138 sigma  0.053495
 Lyr/side  9 1 Chsq  24.63848    8076 sigma  0.055234
 Lyr/side  9 2 Chsq  21.74773    7907 sigma  0.052445
 Lyr/side 10 1 Chsq  21.41700    7587 sigma  0.053131
 Lyr/side 10 2 Chsq  23.84718    8541 sigma  0.052840
 Lyr/side 11 1 Chsq  24.36734    7930 sigma  0.055433
 Lyr/side 11 2 Chsq  24.55533    7862 sigma  0.055886
 Lyr/side 12 1 Chsq  26.76544    7773 sigma  0.058680
 Lyr/side 12 2 Chsq  28.48604    8463 sigma  0.058017
 Lyr/side 13 1 Chsq  26.40359    7115 sigma  0.060918
 Lyr/side 13 2 Chsq  29.64868    7417 sigma  0.063225
 Lyr/side 14 1 Chsq  26.57665    6257 sigma  0.065173
 Lyr/side 14 2 Chsq  29.65096    7058 sigma  0.064815
 Lyr/side 15 1 Chsq  26.65633    5915 sigma  0.067131
```

```
 Lyr/side 15 2 Chsq  31.78499    6854 sigma  0.068099
 Lyr/side 16 1 Chsq  23.19174    5120 sigma  0.067303
 Lyr/side 16 2 Chsq  25.72200    5736 sigma  0.066965
 Lyr/side 17 1 Chsq  23.22954    5227 sigma  0.066664
 Lyr/side 17 2 Chsq  26.77862    5804 sigma  0.067925
 Lyr/side 18 1 Chsq  25.07768    5365 sigma  0.068369
 Lyr/side 18 2 Chsq  25.32584    5621 sigma  0.067124
 Lyr/side 19 1 Chsq  21.23690    4707 sigma  0.067170
 Lyr/side 19 2 Chsq  22.96522    5093 sigma  0.067150
 Lyr/side 20 1 Chsq  19.36103    4734 sigma  0.063951
 Lyr/side 20 2 Chsq  20.00561    5331 sigma  0.061259
 Lyr/side 21 1 Chsq  16.03384    4734 sigma  0.058198
 Lyr/side 21 2 Chsq  16.63572    5303 sigma  0.056009
 Lyr/side 22 1 Chsq  16.67936    4536 sigma  0.060639
 Lyr/side 22 2 Chsq  16.06007    4866 sigma  0.057450
 Lyr/side 23 1 Chsq  17.58371    3682 sigma  0.069106
 Lyr/side 23 2 Chsq  24.46763    4447 sigma  0.074176
```

4.2 Iteration 2

Step 1:
Modify the CARD_CIRC.CRD file to look exactly like the following. One wants to turn off writing of a PDT, and to cause the program to use the calibration table from the previous iteration. On the Alliant FX/8, this reprocessing of the PDT takes about 110 ms per event, (compared to about 230 ms per event to create the PDT).

```
WRIT 4
LIST
FZIN  'XT'  0
USER 6=35000
USER 7=0
XTAL 'NONE'
GLOB 'NONE'
STOP  <===== READING STOPS HERE ========
```

and modify the CARD_CIRC.JDC card file for the following card.

```
ZPRB   0.005
DYMX   0.050
SYDQ   2.00E-4
SXDQ   0.008
```

Now run the CBCIRC code using the PDT that was generated in step one as input. **Step 2:** Repeat steps 2 through 4 of iteration 1.

4.3 Iteration 3

Step 1:
Modify the CARD_CIRC.JDC card file for the following cards.

```
ZPRB   0.005
DYMX   0.030
SYDQ   0.000
SXDQ   0.000
```

Now run the CBCIRC code using the PDT that was generated in step one as input. **Step 2:**
Repeat steps 2 through 4 of iteration 1.
Step 3:
Repeat this iteration a second time.
The following is an example of the FIT.CHI file from the FITRPHI program after the second iteration using this method.

```
Lyr/side  1 1 Chsq   1.08440    6886 sigma  0.012549
Lyr/side  1 2 Chsq   0.86027    5993 sigma  0.011981
Lyr/side  2 1 Chsq   1.00426    7678 sigma  0.011437
Lyr/side  2 2 Chsq   0.94678    7393 sigma  0.011317
Lyr/side  3 1 Chsq   1.18016    8432 sigma  0.011831
Lyr/side  3 2 Chsq   1.00615    8067 sigma  0.011168
Lyr/side  4 1 Chsq   1.05103    7855 sigma  0.011567
Lyr/side  4 2 Chsq   1.02037    8104 sigma  0.011221
Lyr/side  5 1 Chsq   1.14394    8086 sigma  0.011894
Lyr/side  5 2 Chsq   0.99985    7720 sigma  0.011380
Lyr/side  6 1 Chsq   1.11023    7788 sigma  0.011940
Lyr/side  6 2 Chsq   1.11062    8122 sigma  0.011694
Lyr/side  7 1 Chsq   1.14784    8067 sigma  0.011928
Lyr/side  7 2 Chsq   0.99668    7443 sigma  0.011572
Lyr/side  8 1 Chsq   1.06591    7549 sigma  0.011883
Lyr/side  8 2 Chsq   1.07407    7855 sigma  0.011693
Lyr/side  9 1 Chsq   1.02883    7760 sigma  0.011514
Lyr/side  9 2 Chsq   0.95034    7211 sigma  0.011480
Lyr/side 10 1 Chsq   0.96721    7327 sigma  0.011489
Lyr/side 10 2 Chsq   0.95055    7621 sigma  0.011168
Lyr/side 11 1 Chsq   1.03165    7577 sigma  0.011669
Lyr/side 11 2 Chsq   0.85062    7187 sigma  0.010879
Lyr/side 12 1 Chsq   0.95523    7191 sigma  0.011525
Lyr/side 12 2 Chsq   0.88891    7578 sigma  0.010831
Lyr/side 13 1 Chsq   0.98150    7540 sigma  0.011409
Lyr/side 13 2 Chsq   0.82681    7170 sigma  0.010739
Lyr/side 14 1 Chsq   0.96877    6848 sigma  0.011894
Lyr/side 14 2 Chsq   0.94219    7502 sigma  0.011207
Lyr/side 15 1 Chsq   1.02055    7314 sigma  0.011812
Lyr/side 15 2 Chsq   0.91594    7100 sigma  0.011358
Lyr/side 16 1 Chsq   0.89882    5917 sigma  0.012325
Lyr/side 16 2 Chsq   0.92340    6958 sigma  0.011520
Lyr/side 17 1 Chsq   1.06195    6555 sigma  0.012728
Lyr/side 17 2 Chsq   0.82881    6536 sigma  0.011261
Lyr/side 18 1 Chsq   0.88826    6034 sigma  0.012133
Lyr/side 18 2 Chsq   0.88115    6775 sigma  0.011404
Lyr/side 19 1 Chsq   0.81429    5288 sigma  0.012409
Lyr/side 19 2 Chsq   0.66018    5035 sigma  0.011451
```

```
 Lyr/side 20 1 Chsq   0.66415    5082 sigma  0.011432
 Lyr/side 20 2 Chsq   0.63912    5572 sigma  0.010710
 Lyr/side 21 1 Chsq   0.65802    5375 sigma  0.011064
 Lyr/side 21 2 Chsq   0.66837    5346 sigma  0.011181
 Lyr/side 22 1 Chsq   0.68170    5252 sigma  0.011393
 Lyr/side 22 2 Chsq   0.66906    5467 sigma  0.011063
 Lyr/side 23 1 Chsq   0.75858    4717 sigma  0.012681
 Lyr/side 23 2 Chsq   0.65430    5297 sigma  0.011114
```

4.4 Iteration n

This iteration is used to see how good the calibrations are.
Step 1:
Modify the CARD_CIRC.JDC card file to look like the following:

```
WRIT 4
LIST
DEBG   F F F F F F F F F F
NDBG   0

DELZ  23*0.90
ANGL    90.00  1.0
IAMP   400     1
BMAG   -15.    1.0

LGT2   T

DELY  0.0250
DLY2  0.0250
DELT  0.0050

JDCL   T
ICAL   -1
LGWR   0

MCIR   12
ZOFF   0.00  1.00
ZPRB   0.005
CLIM  40.0
CFRA  0.25
CSMX  0.2618
DYMX  0.0300
SY02   2.00E-5
SYDF   5.00E-5
SYDQ   0.000
STOP  <======= READING STOPS HERE ====
   DEBG(1) : Overall debug control
   DEBG(2) : Debug unpacking raw data.
   DEBG(3) : Debug pattern recognition.
   DEBG(4) : Debug circle fits.
```

```
   DEBG(5) : Debug helix fits.
   DEBG(6) : Debug vertex fits.
   DEBG(10): Debug the slow control processing.
```

and then run the CBCIRC code as usual.
Note that no CALCIRCLE.DAT or JDCSLFREF.DAT files will be written. **Step 2:**
Create a version of PLOT_CIRC using the pilot *CIRPLT0, and run the CIRC.HIST file through it. During this pass, make sure to draw the resolution pictures.
Step 3:
Create a version of the PLOT_CIRC code using the *CIRPLT1 pilot, and run the CIRC.HIST file through it. The produced pictures from the previous two iteartions give a measure as to how good the calibrations are.

5 dE/dx calibration of the JDC

The dE/dx calibration consists of one pass through the data using the CBCIRC program generated in the previous section. Use the CARD_CIRC.CRD card file for steering CBCIRC, and modify the CARD_CIRC.JDC file to look like the following, (be sure to remove the comments before using).

```
WRIT 4                ! Echo to unit 4.
LIST                  ! Echo!

DELZ  23*0.90         ! Z-resolution.
ANGL    90.00  1.0    ! Angle offset to sector 1.
IAMP   400     1      ! JDC pulse height cut.
BMAG   -15.    1.0    ! Magnetic field strength.

LGT2   T              !

JDCL   T              ! Do JDC calibration.
ICAL   3              ! Calibration type 3, (dE/dx).
LGWR   1              ! Bad wires in z are not used.

ZOFF   0.00  1.00     ! Offset to target center = 0.0

SY02   2.00E-5        ! Constant term in resolution function.
SYDF   1.00E-4        ! Term proportional to sqrt(distance).

STOP  <======= READING STOPS HERE ====
    DEBG(1) : Overall debug control
    DEBG(2) : Debug unpacking raw data.
    DEBG(3) : Debug pattern recognition.
    DEBG(4) : Debug circle fits.
    DEBG(5) : Debug helix fits.
    DEBG(6) : Debug vertex fits.
    DEBG(10): Debug the slow control processing.
```

Run the CBCIRC program as you normally would. The program will stop by itself after it has obtained enough events, (20000?). The output will be a file NEWGAIN.DAT, which should replace the input gain file for running the program. Questions about this should be addressed to Klaus Peters as I did not write this code.

References

[1] Gunter Folger, CB–Note 121, **Offline Reconstruction Software**.

[2] Curtis A. Meyer, CB–Note 93, **Chamber Reconstruction Software**.

[3] Curtis A. Meyer, CB–Note 123, **Users Guide for Locater**.

[4] Curtis A. Meyer, CB–Note 124, **Z–Calibration of the JDC**.

[5] F. James and M. Roos, **MINUIT, Function Minimization and Error Analysis, Release 89.03**, *CERN Computer Center Program Library Long Write–Up D506*.

[6] R. Brun and D. Lienart, **HBOOK User Guide, Version 4**, *CERN Computer Center Program Library Long Write–Up Y250*

[7] R. Brun and N. Cremel Somon, **HPLOT User Guide, Version 5**, *CERN Computer Center Program Library Long Write–Up Y251*

[8] H. J. Klein and J. Zoll, **PATCHY Reference Manual**, *CERN Program Library*.

[9] M. Brun, R. Brun and F. Rademakers, **A Source Code Management System, CMZ, User's Guide & Reference Manual**.

[10] F. H. Heinsius and T. Kiel, CB–Note 92, **Crystal Data Reconstruction Software**.