AGS CNI Polarimeter Software Description

Jeff Wood

wood@physics.ucla.edu

(310) 794-4259

Outline

- Location of source code and executables
  - Setting system environment
- Brief description of polarimeter commands
  - ADO handler (polado)
  - A simple control script (cnirun.pl)
  - DAQ software (cnipol)
  - α source calibration
  - Deadlayer and t0 fit
  - Analysis software (asym)
- Detailed description of analysis software

Software Location

Computer where software is located:

cnipol.ags.bnl.gov

Username: cniuser

Password: contact J.Wood for password

Location of executable file (for FY05):

$CMDDIR=~cniuser/cni2005-ado/

Directory location of source code (for FY05):

$SRCDIR=~cniuser/cni2005-dev/

All commands can be issued from either $CMDDIR or $SRCDIR

Environment Variables

The following environment variables must be set in order to communicate with the CDEV server. These should be set automatically, but if the programs return errors related to CDEV these variable should be checked.

- CDEV=/usr/local/cdev/store/X86
- CDEVSHOBJ=/usr/local/cdev/store/X86/lib
- CDEVDDL=/usr/local/cdev/AgsPolarClient.ddl
- CDEV_NAME_SERVER=acnsun68.pbn.bnl.gov
- LD_LIBRARY_PATH=/usr/local/cdev/store/X86/lib:/usr/local/isptools/ispvmsystem/lib:/usr/local/root/cern/lib

ADO Handler Software

This program interprets commands from the polarimeter ADO manager used by operators and C-AD physicists. It automatically moves target in/out of beam, starts/stops DAQ, and runs analysis software. Program will start at boot up if last line of /etc/inittab is uncommented.

- To run at command line: cd $CMDDIR
  ./polADO.sh
- Source code: $SRCDIR/polado.cpp
  $CMDDIR/polADOmeasure.sh
  $CMDDIR/polADO.sh
- To compile: cd $SRCDIR
  make
  make install

note: 'make install' copies executables from $SRCDIR to $CMDDIR

Simple Control Script

This Perl script can be used to run the polarimeter instead of the ADO program described on the previous page. The control window has buttons to start/stop the DAQ, start/stop a target scan, and run analysis program. The target must be controlled with the C-AD “pet page”.

- To run: cd $SRCDIR

  ./cnirun.pl &
- Use the “User” menu (upper-right) to select the active PPM User. This must be set to read the AGS intensity from CDEV. This also sets the appropriate versions of the files current_set.ini and asym/asym_cut.dat
- The “Expert” button displays a PS file with many plots used for quality assurance. The “Plot” button opens a file with just the most relevant plots.

DAQ Software

This program runs the DAQ for taking data or doing target scans. Data and log files are written to $SRCDIR/data and $SRCDIR/log. Target scan writes no files.

To run: cd $SRCDIR

cp config/[desired file] current_set.ini

target scan: cnipol/cnipol –d001 –u[user]

data run: cnipol/cnipol –n[run number] –d001 –u[user]

The argument [user] should be the active PPM User for the AGS. This is needed to accurately read the AGS intensity and cycle numbers. If –u[user] is not specified, then the default value is 4.

DAQ Software (continued)

- Source code located in $SRCDIR/cnipol/ :
  bufferfun.c, camacfun.c, cnipol.c, pcicamac.c,
  readCdev.cpp

- Include files located in $SRCDIR/cnipol/ :
  cnidata.h, cnipol.h, pcicamac.h, rcdev.h

- To compile: cd $SRCDIR/cnipol/
  make
  make install

α Source Calibration

Use DAQ program to take data. Make hbook file with Osamu's analysis program. Fit histograms with PAW kumac. Generate calibration file with Perl script. See also the file $SRCDIR/calib/HowToCalibrate.txt

To run: cd $SRCDIR

cnipol/cnipol –n[run number] –d001 –C[time] -i

The argument [time] is the length of the calibration run in sec.

To analyze: cd $SRCDIR/calib/

cat ../data/[run number].data | ./asym_tool –C –o hbook/[run number].hbk

α Source Calibration (continued)

Check that the proper attenuation factor is set in the file param.kumac. Look for the line:

attenuator=0.1995

Run kumac: pawX11

PAW > h/file 1 hbook/[run number].hbk

PAW > exe param#calib

The calibration parameters are written to calib_out.dat :

cp calib_out.dat alpha/calib_[run number].dat

Edit script genParamFile.pl for new calibration data.

Edit line: $escale_file = “alpha/calib_[run number].dat”;

Generate a new parameter file:

./genParamFile.pl

cp paraout.dat ../sicalib.data

Deadlayer & t0 fit

The deadlayer and t0 fit is similar to the α source calibration, and also described in the file $SRCDIR/calib/HowToCalibrate.txt

Take a normal data run:

cd $SRCDIR

cnipol/cnipol –n[run number] –d001

Convert data file to hbook:

cd $SRCDIR/calib/

cat ../data/[run number].data | ./asym_tool –DZ –o hbook/[run number].hbk

Check that the latest α calibration file is set in the dlayer macro in param.kumac. (continued on next page).

Deadlayer & t0 fit (continued)

Look for the line:

v/read strip,cal,cale [filename] MATCH=-/#/

eg. [file name] = alpha/calib_90002.dat

Run kumac: pawX11

PAW > h/file 1 hbook/[run number].hbk

PAW > exe param#dlayer

The calibration parameters are written to dlayer_out.dat :

cp dlayer_out.dat dlayer/dlayer_t0_[run number].dat

Edit script genParamFile.pl for new deadlayer & t0 data.

Edit line: $dlayer_file = “dlayer/dlayer_t0_[run number].dat”;

Generate a new parameter file:

./genParamFile.pl

cp paraout.dat ../sicalib.data

Analysis Software

Open data file. Loop over all events. Impose event selection and calculate asymmetries. Send results to CDEV server. Log files are written to $SRCDIR/asym/log/.

To run: cd $SRCDIR

cp asym/cuts/[desired file] asym/asym_cut.dat

asym/asym –n[run number]

To view plots: gv asym/plots/asym_[run number].ps &

Event selection criteria are contained in the asym_cut.dat file. Additional selection criteria can be specified by passing arguments to the ‘asym/asym’ command. Type ‘asym/asym –h’ for help.

Description of source code files

- asymdata.c – Main program. It opens the data file, which contains records for beginning/end of each spill (*i.e.* AGS cycle), each detected event, and the beginning/end of the run. The program calls Fortran subroutines for each of these records in the data file.
- asym.F – Contains following subroutines:

  asyminit – Initializes global event counters. Opens log file. Reads event cut parameters from file. Energy (-t) bins are defined.

  asymbeg – Initializes event counters for each spill.

  asym – Imposes event selection by calling evsel subroutine. Sums event counters for each spill. Also, calls fillspmass subroutine contained in hbook.F

  asymend – Sums global event counters. Calculates spill by spill asymmetries by calling sqrtasym. Also, calls hbook.F subroutines fillasym and fillchisq.

  asymendrun – Calculates the mean analyzing power. Calculates final asymmetries by calling sqrtasym subroutine.

- hbook.F – Contains following subroutines:
  - bookhbk – Opens hbook file and books all histograms.
  - fillhbkev – Fills many histograms such as strip dist. and “banana” plots.
  - fillspmass – Fills spin-sorted mass histos for each strip. Used for offline analysis.
  - fillasym – Fills several asymmetry vs. spill histograms
  - fillchisq – Fills asymmetry histos for every 8^{th} spill, used to calculate χ^2.
  - closehbk – Closes hbook file.
  - fillcuts – Not currently used.
  - bookscl and fillhbksc – Used only for scalar mode operation. Scalar mode operation is not yet used. Further testing/debugging needed.
- writeCdev.cpp – Sends calculated asymmetries and image file to the CDEV server.
- asymramp.F – Essentially the same as asym.F, but all variables have an index for each GCC bin during the ramp.
- asymplot.kumac – PAW macro used to generate an image file from the histograms generated by hbook.F.

Include files

The include files are used to declare variables, common blocks, and data structures that are used by analysis routines. All include files are contained in the directory $SRCDIR/asym/inc/

- asyms.h – All asymmetry variables declared here.
- config.h – Some variables from the configuration data record.
- cuts.h – Declare variables that are read from asym_cut.dat and used for event selection.
- ebins.h – Variables used to define the energy binning.
- event.h – Variables for each event, *e.g.* energy, tof
- eventcnt.h – All of the event counter variables.

Include files (cont.)

- cntramp.h – Counter variables used for ramp measurements. Each variable is an array with length equal to the number of GCC bins.
- scaler.h – Variables used for scaler mode only.
- scalcnt.h – Counter variables used for scaler mode only.
- cdevdat.h – Declares data structures that are used in asymdata.c and writeCdev.cpp. If changes are made to event.h, scaler.h, config.h, or asyms.h, then the data stuctures should be changed in cdevdat.h too.

Cut parameters

asym_cut.dat – contains parameters used for event selection.

These are the files and routines used for “normal” event mode running. Additional subroutines are called for ramp mode or scaler mode.

asymdata.c
open data file
call asyminit
call bookhbk
search for data records, for loop {
beg spill: call asymbeg
end spill: call asymend
each event: call asym
call fillhbkev
}
call closehbk
call asymendrun
execute asymplot.kumac
call setLogInfo
close data file

asym.F
asyminit
asymbeg
asym: call evsel
call fillspmass
asymend: call sqrtasym
call fillasym
call fillchisq
asymendrun: call sqrtasym
evsel
sqrtasym

hbook.F
bookhbk
fillhbkev
fillasym
fillchisq
closehbk

writeCdev.cpp
setLogInfo