

CLAS-NOTE-94-003

**SDA\_Ec 3.1**  
FORWARD ELECTROMAGNETIC CALORIMETER  
RECONSTRUCTION FOR SDA  
K.B.BEARD  
BEARD@CEBAF.GOV  
FEBRUARY 25, 1994

SDA\_Ec 3.1 uses the Ec3.1.3 release of the CLAS forward electromagnetic calorimeter (Ec) event reconstruction software. It is intended to make Ec3.1.3 simple to place into SDA.

## 1 Introduction

To make the use of Ec <sup>1</sup> as simple as possible by the SDA <sup>2</sup> code, SDA\_Ec reduces everything to one call:

```
call SDA_Ec_recon(Xo,Xd,idx,buffer,ptrs,MAXtrk,Ntrk,trks,sectors,OK,err)
```

SDA is expected to provide a plane is specified by a point  $Xo(\mathcal{B})$  and normal  $Xn(\mathcal{B})$ ; all the reconstructed tracks are then extrapolated to this plane. SDA must also provide the data ( $idx$ ,  $buffer$ , and  $ptrs$ ) as recovered from *EvGen\_get\_event*.  $MAXtrk$  sets an upper limit on the number of  $trks$  to be returned.

All initialization occurs on the first call and takes considerable time to complete.

The routine returns the number of tracks found  $Ntrk$ , the sector the each track which was found in  $sectors$ , and puts the tracks sequentially into array  $trks$ .

## 2 Coordinate System

The Ec package is reports its results in the standard CLAS XYZ coordinate system, while SDA prefers the sector based S123 system. The CLAS Z is parallel to the beam, Y is straight away from the earth, and X is horizontal through the center of sector 1. In SDA's S123 system, S1 is parallel to the beam axis, S2 is through the center of the sector, and  $S3=S1 \times S2$ . All values input to and reported by this routine are in the S123 system.

## 3 SDA track

The SDA package currently assumes tracks are described by ten numbers; a S123 position, a direction, a momentum, a mass, a charge, and the distance along the path from the target. The  $trk$  vector consists of these ten values, repeating the order for each track.

---

<sup>1</sup>Ec 3.1.3, Forward Electromagnetic Calorimeter Reconstruction Software, CLAS-note-94-002, K.B.Beard, February 23, 1994

<sup>2</sup>sTANDARD dATA aNALYSIS, CEBAF, B.Niczyporuk, unpublished

- 1 SDA-X [S123] position on plane
- 2     Y
- 3     Z
- 4 SDA-X [S123] direction
- 5     Y
- 6     Z
- 7 SDA momentum [GeV/c]
- 8 SDA mass [GeV/c<sup>2</sup>]
- 9 SDA charge [|e|]
- 10 SDA path [cm]

The charge, path, and mass cannot be determined by  $E_c$  alone, and are left unchanged. The energy of the shower is put into the SDA momentum.

### 3.1 Terminology and Units

All quantities returned are in GeV, cm, or elemental charges.

### 3.2 Error Messages

The  $E_c$  package is still being developed; in particular the reconstruction algorithms are being improved and made more robust and faster.  $E_c3.1.3$  carries a considerable burden of error checking; as the correctness of the code is verified these routines will gradually be removed in newer releases. The *OK* status returned by most subroutines and the *err* messages are primarily to detect flawed code; the path to the error as well as the error is returned. In general, correct code should never produce an *OK=.FALSE.* status. The character string *err* should be declared sufficiently long to avoid truncating error messages (typically *err* is declared to be 800 bytes long).

## 4 Data Input

The only supported data input are CODA-CLAS 2.0 buffers with the appropriate, filled in, index and pointer table.

## 5 Graphics

The Ec package contains HIGZ graphics intended to minimize the interaction between any host and Ec. The Ec graphics are scaled to the physical dimensions and intended to be superimposed onto an existing HIGZ window.

## 6 Acknowledgements

The author wishes to make clear that no Hampton University resources, time, or supplies were used on this project, and that no responsibility, encouragement, nor support of any kind should be attributed to the University with regards to this project. The author does wish to thank L.Dennis, T.Y.Tung, and M.Guidal for their support.

## A General Ec Routines

The only routines the SDA\_Ec user need call are described briefly here. Results can be accessed using alternative methods; only a small subset of the information is available through the arguments.

*Subroutine* SDA\_Ec\_recon ( X, n, idx, buffer, ptrs, MAXtrk, Ntrk, trks, sectors, OK, err)

*Real* X(3), n(3), trks(\*)

*Integer* idx, buffer(\*), ptrs(\*), MAXtrks, Ntrks, sectors(\*)

*Logical* OK

*Character*\*(\*) err

This subroutine will initialize Ec once. Each time it is called, it will unpack the CODA-CLAS 2.0 format event *buffer* using the beginning of data pointer *idx* and a previously filled lookup table *ptrs*. *MAXtrk* is used to set an upper limit on the number of tracks to packed into *trks*. The results of the reconstruction are SDA-type tracks, extrapolated to a plane defined in the sector based [S123] coordinate system by a point on the plane  $X(1), X(2), X(3)$  and a normal  $n(1), n(2), n(3)$ . The structure of *trks* repeats for each track, the number of tracks present is returned as *Ntrks*. For each track there is its corresponding *sectors* ID number. This routine will analyze everything it can, and report serious errors to *OK* and *err*.

*Subroutine* SDA\_Ec\_CLASCODA ( command, idx, buffer, ptrs, OK, err)

*Character*\*(\*) command, err

*Integer* idx, buffer(\*), ptrs(\*)

*Logical* OK

This routine is for temporary storage of the quantities containing and describing a CLAS-CODA 2.0 format data buffer. The *commands* are "CLEAR", "KEEP", and "RECOVER". The first zeros *idx*, *buffer*, and *ptrs* and marks the internal copy as invalid; the second makes an internal copy of three, the last copies the internal copy into the three. The routine need not be cleared before new data is kept. It is intended to simplify moving the CLAS-CODA buffer between SDA routines.

## B Example Shell

Intended as an example shell, SDA\_Ec\_engine.exe is a stand alone program.

## C Example Session

```
cebafh> ../bin/SDA_Ec_engine.exe
```

```
SDA_Ec_engine.....K.B.Beard 2/94
```

```
Simple analysis engine for using the Ec3.1 package  
using CLAS 2.0 format
```

```
>>>for testing of Ec package- type /HELP for help
```

```
List of valid workstation types:
```

```
0: Alphanumeric terminal  
1-10: Describe in file higz_windows.dat  
n.host: Open the display on host (1 < n < 10)  
m: PAW_MOTIF on local host  
m.host: PAW_MOTIF on specified host  
7878: FALCO terminal  
7879: xterm
```

```
Metafile workstation types:
```

```
-111: HIGZ/PostScript (Portrait)  
-112: HIGZ/PostScript (Landscape)  
-113: HIGZ/Encapsulated PostScript  
-777/8: HIGZ/LaTeX
```

```
workstation type?
```

```
1 !PAW workstation type 0=nothing
```

```
CLAS 2.0 format event file?:  
../dat/e.evt      !CLAS-CODA2.0 format event file
```

```
ECENGINE> /help
```

```
EcEngine>      analysis shell  
/HElp          list options  
/EXIT          exit properly  
/ABORT         stop suddenly  
/MAXimum:n     maximum number of objects/event  
/TECHnique:mthd analysis method  
/Nevents:n     number events to process  
/ERRor:op      error action [HALT,CONT]  
/PLOT:op       plotting [ERR,ALL]  
/SECTors:m:n   selected sectors  
/STATus        show status  
EC             enter Ec interface  
      cmdnd    give Ec command
```

```
ECENGINE> /plot:ALL /Nevents:1  
Nevents to process: 1  
ECENGINE> %
```

```
begin Ec_initialize_all.....  
version: January 1994 Ec3.1/PIXELS/NO_HOST/SIM/GRAPHICS  
...Ec_init done...
```

Ntrk= 1

```
track# 1  sector= 1
trk([S123] position(X) cm      )= 283.1855
trk([S123] position(Y) cm      )= 110.0097
trk([S123] position(Z) cm      )= 14.29458
trk([S123] direction(X)        )= .9548206
trk([S123] direction(Y)        )= .2954937
trk([S123] direction(Z)        )= 3.16395E-02
trk(momentum [GeV]             )= .8517242
trk(mass [GeV]                  )= .0
trk(charge |e|                  )= .0
trk(path from target cm        )= .0
```

.....

```
1 events just processed
1 events analyzed total
ECENGINE> /abort
```

## D Bugs

The HIGZ routines seem to have problems opening an Xwindow (workstation type 1) on a remote host, even if the environmental DISPLAY variable is set properly. If PAW is run first in that session and used to open a window, then exited, the HIGZ routines can open the window again. For example, on the Xwindow terminal ccxt6.ceba.gov:

```
setenv DISPLAY ccxt6.ceba.gov:0.0
paw
....
Workstation type (?=HELP) <CR>=1 : 1.ccxt6.ceba.gov
....
PAW > quit
```

will work around the problem.

## **E Linking**

The makefile **makefile.SDA\_Ec** is used both for **ULTRIX** and **HP-UX**; all the machine dependence is buried in the **CLASsoft\_definitions.ULTRIX** and **CLASsoft\_definitions.HP-UX** files. **Temporarily** the code is on CEBAF's **/usr/user4/beard/CLASsoft/SDA\_EC/\*** directories.