



3^{er} Taller de Cómputo de la Colaboración MexNICA

Reconstrucción en el marco de trabajo del MpdROOT

J. Maldonado¹, Dra. I. Maldonado¹, Dra. I. Domínguez¹, Dra. Ma. E.
Tejeda-Yeomans², Dra. G. Espinoza³

¹ Facultad de Ciencias Físico Matemáticas, Universidad Autónoma de Sinaloa, ² Facultad de Ciencias -
CUICBAS, Universidad de Colima, ³ BUAP

February 2, 2021

Outline

Reconstrucción

Reconstrucción

*"La reconstrucción de eventos es el proceso de interpretar las señales electrónicas producidas por el detector para determinar las partículas originales que pasaron, sus momentos, direcciones y el vértice principal del evento."*¹

Cadena de reconstrucción:

- ▶ Reconstrucción de hits en los detectores
- ▶ Reconstrucción de trazas
 - ▶ Búsqueda de trazas en el rastreador principal.
 - ▶ Propagación de trazas, por ejemplo con el filtro de Kalman.
 - ▶ Coincidencia con otros detectores (Trazas globales).
- ▶ Hallazgo del vértice
- ▶ Identificación de partículas

¹<http://mpd.jinr.ru/mpd-event-reconstruction>

Macro reco.C

Una vez llevada a cabo la simulación de eventos, con algún generador y la simulación del detector con el macro runMC.C, debemos llevar a cabo el proceso de reconstrucción, lo cual lo hacemos con el macro **reco.C**

- ▶ Cargamos mpdroot y copiamos el archivo reco.C a nuestro directorio de trabajo

```
source ~/Software/mpdroot/build/config.sh  
cp ~/Software/mpdroot/macro/mpd/reco.C .
```

- ▶ Revisar que en el archivo mpdloadlibs.C se cambie:
libMCStack → libMpdMCStack

Argumentos para llevar a cabo la reconstrucción:

- ▶ Archivo de entrada de la simulación: **evetest.root**
- ▶ Archivo de salida de la reconstrucción: **mpddst.root**
- ▶ Número de evento a partir del cual se inicia la reconstrucción: por defecto **0**
- ▶ Número de eventos a analizar: **20** en este caso
- ▶ Tipo de análisis: **local**

```
void reco(TString inFile = "evetest.root", TString outFile = "mpddst.root",  
Int_t nStartEvent = 0, Int_t nEvents = 20, TString run_type = "local")
```

```
reco.C (~Software/Taller2/reconstruccion) - GVIM6
File Edit Tools Syntax Buffers Window Help
#include "MpdEtofMatching.h"
#include "MpdFillDstTask.h"
#include "MpdGetNumEvents.h"
#include "MpdEmcHitCreation.h"

#include <iostream>
using namespace std;
#endif

R_ADD_INCLUDE_PATH($VMCWORKDIR)
#include "macro/mpd/mpdloadlibs.C"

#define UseMlem // Choose: UseMlem HitProducer

// Macro for running reconstruction:
// inFile - input file with MC data, default: evetest.root
// nStartEvent - number (start with zero) of first event to process, default: 0
// nEvents - number of events to process, 0 - all events of given file will be processed, default: 1
// outFile - output file with reconstructed data, default: mpddst.root
// run_type - proof execution, default "local". e.g.:
// "proof" - run on proof-lite with "CPU" count workers,
// "proof:workers=3" - run on proof-lite with 3 workers
// "proof:user@proof.server:21001" - to run on the PROOF cluster created with PoD (under user 'MPD', default port - 21001)
// "proof:user@proof.server:21001:workers=10" - to run on the PROOF cluster created with PoD with 10 workers (under USER, default port - 21001)
// nc-farm : proof:mpd@nc10.jinr.ru:21001
void reco(TString inFile = "$VMCWORKDIR/macro/mpd/evetest.root", TString outFile = "mpddst.root", Int_t nStartEvent = 0, Int_t nEvents = 10, TString run_type = "local") {
    // =====
    // Verbosity level (0=quiet, 1=event level, 2=track level, 3=debug)
    Int_t iVerbose = 0;

```

42,0-1

17%

Los elementos más relevantes en la reconstrucción son:

- ▶ FairRunAna* fRun → maneja el proceso de reconstrucción.
- ▶ Configura el archivo de entrada con datos o experimentales y el archivo de salida en formato DST.

```
FairSource* fFileSource = new FairFileSource(inFile);  
fRun->SetSource(fFileSource);  
fRun->SetOutputFile(outFile);
```

- ▶ Añade tareas de análisis y permite configurar el proceso de reconstrucción

```
MpdKalmanFilter *kalman = MpdKalmanFilter::Instance("KF");  
fRun->AddTask(kalman);
```

```
FairTask* findVtx = new MpdKfPrimaryVertexFinder("Vertex finder");  
fRun->AddTask(findVtx);
```

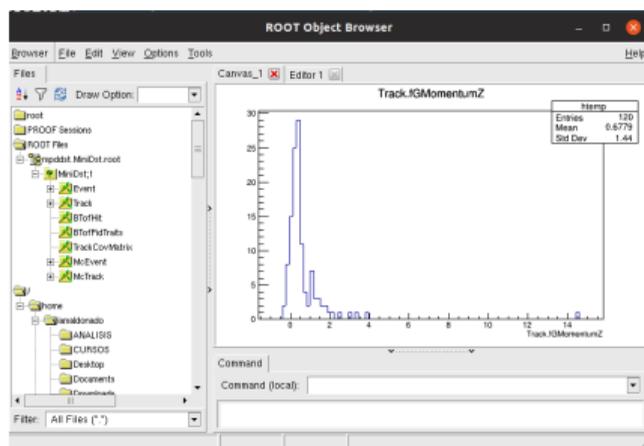
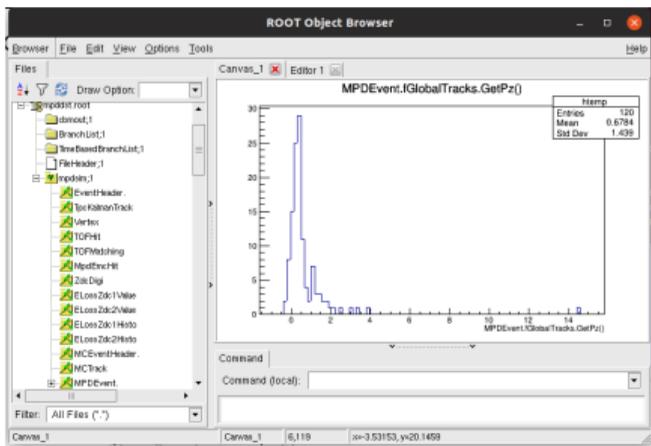
- ▶ Inicializa la tarea y sus variables

```
fRun->Init();
```

Archivo mpddst.root y mpddst.Minidst.root

Datos reconstruidos con/sin información condensada

```
MpdMiniDstFillTask* miniDst = new MpdMiniDstFillTask(outFile);  
fRun->AddTask(miniDst);
```



Corrección de DCA

Las trazas reconstruidos contienen información sobre la distancia de máximo acercamiento, el cual requiere una corrección para la selección de las trazas primarias, para ello se tiene que ajustar los valores de la helicidad

- ▶ Copiamos los archivos rootlogon.C y restore_dca.c a nuestro directorio de trabajo

```
cp ~/mpdroot/macro/physical_analysis/Flow/restore_dca/rootlogon.C .  
cp ~/mpdroot/macro/physical_analysis/Flow/restore_dca/restore_dca.c .  
cp ~/mpdroot/macro/physical_analysis/Flow/restore_dca/restore_dca.h .
```

- ▶ Editamos el archivo rootlogon.C

```
"-I${Boost_INCLUDE_DIRS} " =====>>>> "-I${FAIRROOTPATH}/include ";
```

- ▶ Y realizamos los siguientes pasos

```
root -b  
root [0] gROOT->LoadMacro("../mpdloadlibs.C")  
mpdloadlibs();  
.L restore_dca.c+  
restore_dca("mpddst.root", "mpddst2.root")
```

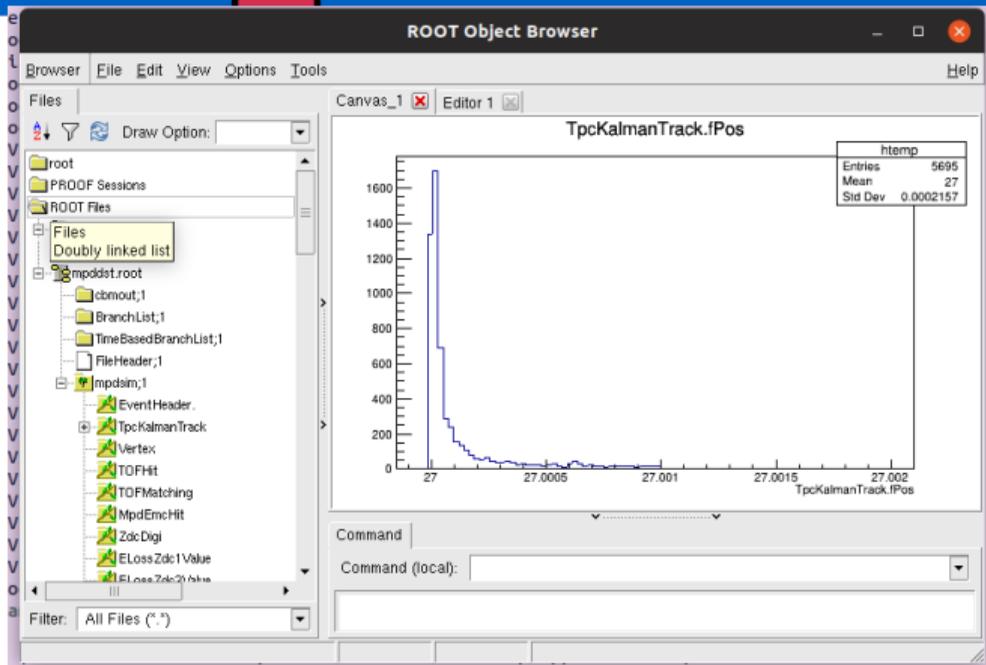
- ▶ y nos da el archivo con la corrección mpddst2.root

```
lamaldonado@supernova:~/Software/Taller2/correction$ root -b
```

```
-----  
| Welcome to ROOT 6.16/00                https://root.cern |  
|                                     (c) 1995-2018, The ROOT Team |  
| Built for linuxx8664gcc on Feb 24 2020, 16:53:00 |  
| From tags/v6-16-00@v6-16-00 |  
| Try '.help', '.demo', '.license', '.credits', '.quit!/.q' |  
-----
```

```
Setting include path ...
```

```
root [0] gROOT->LoadMacro("../mpdloadlibs.C")  
(int) 0  
root [1] mpdloadlibs();  
root [2] .L restore_dca.c+  
root [3] restore_dca("mpdst.root", "mpdst2.root")  
EVENT N 0  
EVENT N 1  
EVENT N 2  
EVENT N 3  
EVENT N 4  
EVENT N 5  
EVENT N 6  
EVENT N 7  
EVENT N 8  
EVENT N 9  
EVENT N 10  
EVENT N 11  
EVENT N 12  
EVENT N 13  
EVENT N 14  
EVENT N 15  
EVENT N 16  
EVENT N 17  
EVENT N 18  
EVENT N 19  
root [4] █
```



Más archivos de calibración para el análisis de Flujo, están descritos en:

[https:](https://git.jinr.ru/nica/mpdroot/tree/dev/macro/physical_analysis/Flow)

[//git.jinr.ru/nica/mpdroot/tree/dev/macro/physical_analysis/Flow](https://git.jinr.ru/nica/mpdroot/tree/dev/macro/physical_analysis/Flow)

Manos a la obra

Explora los archivos de salida

```
root mpddst2.root  
root[0] TBrowser n
```