Table of Contents

Stand-Alone Muon Reconstruction

Complete:

Goals of this page

This tutorial shows how to run the reconstruction of muon tracks in the CMS muon spectrometer (muon system alone) and to analyze its output.

Contacts

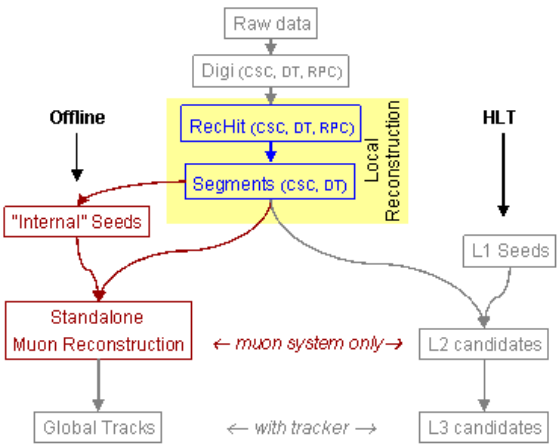
Riccardo Bellan (Riccardo.Bellan@cernNOSPAMPLEASE.ch)

Introduction

The muon reconstruction chain starts with the "local reconstruction", i.e. the reconstruction of hits in DTs, CSCs and RPCs. Hits within each DT and CSC chamber are then matched to form "segments" (track stubs).

In the offline reconstruction, the segments reconstructed in the muon chambers are used to generate "seeds" consisting of position and direction vectors and an estimate of the muon transverse momentum. These initial estimates are used as seeds for the track fits in the muon system, which are performed using segments and hits from DTs, CSCs and RPCs and are based on the Kalman filter technique. The result is a collection of reco::Track objects reconstructed in the muon spectrometer, which are referred to as **"standalone muons"**. To improve the momentum resolution, a beam-spot constraint can be applied in the fit. Two collections of "standalone muons", with and without the beam-spot constraint, are available in the event record (in both RECO and AOD). A more detailed description of the reconstruction of tracks in the muon system alone can be found in Section 4 of CMS AN 2008/097.

Standalone muon tracks are then matched with tracker tracks to form "global muons". A schematic view of the steps of muon reconstruction is shown below.



The local reconstruction chain and all details of the production of DT and CSC segments are presented in the tutorial on local muon reconstruction. In the present tutorial we focus on the standalone muon track reconstruction, starting from a data file containing already reconstructed segments.

Set up your environment

Set your runtime environment (shown for release %WBRELEASENEW%):

```
cd CMSSW_%WBRELEASENEW%/src
cmsenv
```

- Full instructions
- Summary for every login session
- Access modules from the CVS repository.

You don't need to checkout and compile any package to run the reconstruction, you simply need an input sample and a configuration file.

Samples

You can use your favorite already-available sample, or generate one from scratch. The output of the tutorial on local muon reconstruction can be used as input for this step.

For this tutorial we assume that the sample used in the following already contains:

- the DT segments
- the CSC segments
- the RPC rechits.

Configuration file

Checkout the following package:

```
addpkg RecoMuon/StandAloneMuonProducer
cvs co -r V01-00-05-04 RecoMuon/StandAloneMuonProducer/test/StandAloneMuon_cfg.py
```

This package contains the blocks of configuration files and sequences needed to run the standalone muon reconstruction. These files contain default values for the parameters, and can be included in user's configuration files. The user is not supposed to modify these files directly, but can override a card modifying the values of parameters in the cfg file.

- `python/` directory: contains `cff` files that you can include in your `cfg` file to run the standalone muon reconstruction;
- `test/` directory: provides some `cfg` files as example.

In the following we will use the `StandAloneMuon_cfg.py` configuration file.

The Production Chain

In general, to run the Standalone Muon production starting from the output of the local muon reconstruction the cfg file needs to contain:

- the `Source` block to specify a file containing DT, CSC and RPC segments and rechits
- the magnetic field and geometry `cfi` files, and some services
- the module blocks included as `cfi` files for the actual reconstruction:
    - the Muon Seed producer module (`MuonSeed`)

    - the Standalone Muon Track producer module (`standAloneMuons`)
- the `OutputModule` to define an output file
- the `Path` defining the sequence of modules to be executed
    - `process.p = cms.Path(process.MuonSeed * process.standAloneMuons)`

You can get the configuration files for these modules with the simple include statement:

```
process.load("RecoMuon.Configuration.RecoMuon_cff")
```

You may also want to add one or more analyzer modules. To do this you have to specify them and to add them at the end of the execution path.

Run an Example Production Job

In the following we will run the example job:

```
RecoMuon/StandAloneMuonProducer/test/StandAloneMuon_cfg.py
```

which produces the Muon Seeds and Standalone Muon Tracks.

To run the job you must:

1. Modify the `StandAloneMuon_cfg.py` file in order to use your sample and decide where to write the output file:
    1. Modify the `Source` section to point to your favorite sample
    2. Modify the `OutputModule`: `fileName = cms.untracked.string('/tmp/RecoSTAMuons.root')`
2. Run the job: `cmsRun -p StandAloneMuon_cfg.py > & ! $SCRATCH/StandAloneMuon.out &`

At the end of the job the `/tmp/StandAloneMuon.root` output file will contain all the objects produced by the standalone muon reconstruction. You can then analyze it either using a CMSSW analyzer or browsing it with a ROOT TBrowser.

Run An Example Analyzer

We also add a simple analyzer to fill a histogram of the p_T resolution of the muons. It is implemented in: `RecoMuon/StandAloneMuonProducer/test/` STAMuonAnalyzer.h and STAMuonAnalyzer.cc.

Note: you can have a look at the code to find an example about how to access the collection of muon `reco::Track` in the event, loop on it and extract relevant information.

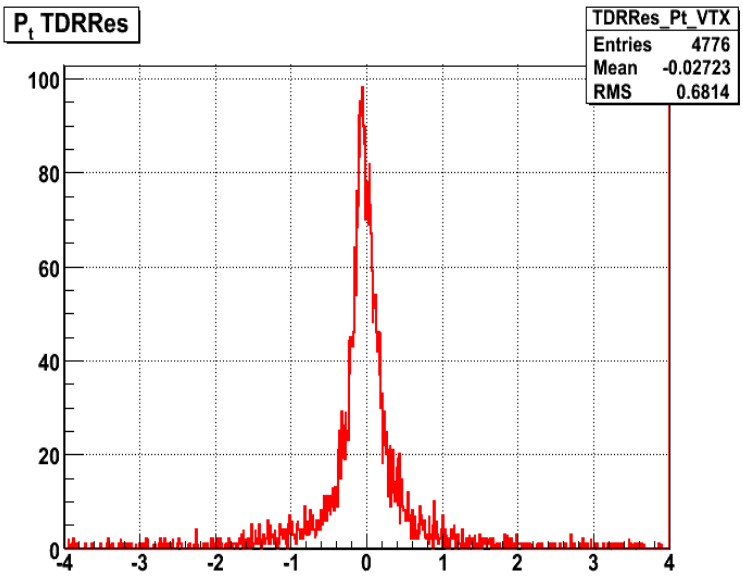
In the configuration file, the analyzer is specified as:

```
process.STAMuonAnalyzer = cms.EDAnalyzer("STAMuonAnalyzer",
                                         DataType = cms.untracked.string('SimData'),
                                         StandAloneTrackCollectionLabel = cms.untracked.string('s
                                         MuonSeedCollectionLabel = cms.untracked.string('MuonSeed
                                         rootFileName = cms.untracked.string('STAMuonAnalyzer.roo
                                         )
```

You can add it to the `path` defined in the production cfg file to combine reconstruction and histogramming tasks in the same job. **Note:** you will find both the analyzer module and the corresponding path *commented* in `StandAloneMuon_cfg.py`. You only need to uncomment them (and comment the default path) to have the

analyzer run.

The analyzer produces the output file STAMuonAnalyzer.root containing the p_T and $1/p_T$ resolution plots.



Review status

Reviewer/Editor and Date (copy from screen)	Comments
RiccardoBellan - 30 Nov 2006	Added initial info
GianlucaCerminara - 09 Feb 2007	Update to CMSSW_1_2_2
JennyWilliams - 23 Mar 2007	reinstated manual contents to enable pdf generation
DanieleTrocino - 24 Feb 2009	Update to CMSSW_2_1_12, cfg moved to StandAloneMuonProducer package

Responsible: RiccardoBellan, Nicola Amapane
Last reviewed by: SlavaValuev - 04 Mar 2009

This topic: CMSPublic > SWGuideStandAloneMuonReco
Topic revision: r29 - 2009-03-04 - SlavaValuev