

# Cuts Software

---

## MICE Analysis Workshop

**Ajit Kurup**

**12<sup>th</sup> April 2018**

**Imperial College  
London**

# Introduction

---

- First version of the cuts software written by Misha Fedorov.
  - Implements 12 cuts based on TOF, momentum, mass, etc.
  - Pass/fail information stored in the recon event.
- Update this to
  - Provide additional info stored with the cut to allow analysing the effect of the cut without having to reprocess.
  - Allow users to easily implement their own cuts.
- Created a new mapper and class to be stored in the recon event.
  - New code structure provides more flexibility and is easier for users to add their own cuts.

# Cuts

- Mapper
  - MapCppCuts implements all the cuts.
- DataStructure
  - Cuts class that has a fixed length vector of bools to store pass/fail information.
  - A Cuts object is stored in ReconEvent.
- Existing cuts:
  - Number of TOF0 hits == 1.
  - Number of TOF1 hits == 1.
  - $28 < \Delta t$  between TOF0 and TOF1 spacepoints  $\leq 32$ .
  - Number of SciFi tracks == 1.
  - Number of SciFiHelicalPRTracks == 1 && number of TKU hits == 5.
  - $130 < P_{TKU} \leq 150$ .

where  $P_{TKU}$  is defined at plane 2, station 5.

  - momentum loss at TKU cut.
    - $(P_{TKU} + 5)/m < \beta\gamma_{TOF} \leq (P_{TKU} + 43.4)/m$   
where  $\beta_{TOF} = dt_e/dt$ .
  - P-value cut
  - Mass cut
    - $99 < (P_{TKU} + 18.82) / \beta\gamma_{TOF} \leq 110$
  - GoodParticle == true

where GoodParticle = AND of all above cuts.

  - Number of SciFiHelicalPRTracks == 1 && number of TKD hits == 5.
  - $50 \leq P_{TKD}$  momentum  $\leq 100$ .

where  $P_{TKD}$  is defined at plane 2, station 5.

# ReconCuts

---

- New mapper module MapCppReconCuts.
  - Base class MapCppReconCutBase used to define cuts.
    - The constructor is passed a Json object that contains cut parameters from the config, e.g. min and max parameters.
    - Member function DoCuts defines the cuts.
  - MapCppReconCuts class defines a vector of MapCppReconCutBase pointers called `_cuts_list`.
  - `_process` method in MapCppReconCuts iterates over `_cuts_list` and calls `DoCuts`.
  - `DoCuts` can contain multiple cuts.
    - Makes computing similar cuts more efficient.
    - For each cut, information can be stored in `ReconEvent` by creating a `ReconCutBase` object.

# ReconCuts

---

- New ReconCutBase class in DataStructure.
  - Stores info about each cut
    - Name – short name for the cut
    - Description – descriptive string, e.g. "TOF0SpacePoints==1"
    - CutPass – Boolean value of the cut for a particular event.
    - CutValue – Value of the cut parameter for a particular event.
  - ReconEvent contains a vector of ReconCutBase pointers.
    - Each cut adds a ReconCutBase object to this vector in.
- Default behaviour if cut cannot be computed.
  - CutValue=false
  - CutValue=unphysical default value.

# ReconCuts

- Implement the cuts from MapCppCuts in the new structure.
  - Cuts grouped into 3 new classes: MapCppReconTOFCuts, MapCppReconTrackerCuts, MapCppReconMiscCuts.
  - Ordering in `_cuts_list` is important as MapCppReconMiscCuts defines GoodParticle cut, which is dependent on the output of the other cuts.
  - Parameters also grouped in the config file.

```
recon_cuts={
  "TOF_cuts":{
    "dt_min":28.,
    "dt_max":32.
  },
  "Tracker_cuts":{
    "min_p_value":0.02,
    "TKU_min_mom":130.,
    "TKU_max_mom":150.,
    "TKD_min_mom":50.,
    "TKD_max_mom":100.,
    "min_mom_loss":5,
    "max_mom_loss":43.4,
    "mom_correction":18.82,
    "min_mass":99.,
    "max_mass":110.
  },
  "Misc_cuts":{
    "good_particle":["TOF0","TOF1","TOFD0","Single_track","TKU_hits",
"TKU_mom_cut","Mom_loss_cut","Mass_cut","Track_p_value"]
  }
}
```

- Cut calculations same as in MapCppCuts except for a few bug fixes.

# How to define your own cuts

- Create a new cut class in MapCppReconCuts
  - e.g. copy MapCppReconTOFCuts.cc to MapCppReconExampleCuts.cc. Similarly for header file.
  - Replace all occurrences of “ReconTOFCuts” with “ReconExampleCuts” in MapCppReconExampleCuts.{cc, hh}
- Edit the constructor of MapCppReconExampleCuts to parse parameters from the config file and define defaults.

```
MapCppReconTOFCuts::MapCppReconTOFCuts(Json::Value cutParams){
    if (!cutParams.isNull()){
        SetMin(cutParams["dt_min"].asFloat());
        SetMax(cutParams["dt_max"].asFloat());
    }else{
        SetMin(-FLT_MAX);
        SetMax(FLT_MAX);
    }
}
```

# How to define your own cuts

- Edit MapCppReconExampleCuts::DoCuts to implement your cuts.
  - Create ReconCutBase object for each cut and add to the ReconEvent

```
// TOF0 cut
if (SpacePoint.GetTOF0SpacePointArray().size() == 1) {
    cutPassed=true;
}else{
    cutPassed=false;
}

anEvent->GetCutsList()->push_back(new ReconCutBase("TOF0",
    "TOF0SpacePoints==1",cutPassed,SpacePoint.GetTOF0SpacePointArray().size()));
```

```
std::stringstream cutDesc;
cutDesc << "TOFDT>=" << GetMin() << " && TOFDT <=" << GetMax();
anEvent->GetCutsList()->push_back(new ReconCutBase("TOFDT",cutDesc.str(),
    cutPassed,dt));
```



# How to define your own cuts

- Edit MapCppReconCuts::\_birth method to include your cut class.

```
// Initialise cuts from cuts_parameters
Json::Value tof_config;

// TOF cuts
try{
    tof_config=JsonWrapper::GetProperty(_set_Cut_params, "TOF_cuts", JsonWrapper::objectValue);
}
catch (const MAUS::Exceptions::Exception& e){
    Squeak::mout(Squeak::warning) << "MapCppReconCuts::_birth Error finding TOF_cuts" << std::endl;
}
MapCppReconTOFCuts* tof_cuts= new MapCppReconTOFCuts(tof_config);
_cuts_list->push_back(tof_cuts);
```

- Update config file with your parameters.

```
recon_cuts={
  "TOF_cuts":{
    "dt_min":28.,
    "dt_max":32.
  },
  "Tracker_cuts":{
    "min_p_value":0.02,
    "TKU_min_mom":130.,
    "TKU_max_mom":150.,
    "TKD_min_mom":50.,
    "TKD_max_mom":100.
```

# Summary and points for discussion

---

- New cuts software ready.
- Hopefully will be merged into MAUS soon.
- Adding new cuts is relatively straight-forward 😊.
- Points for discussion.
  - What cuts to implement next.
  - One definition for “standard” cuts.
  - Separate cuts class for each analysis?
    - Transparency and book-keeping.

# Other cuts to implement

- List from Chris.
  - Number of clusters in TKU track; reject event if all TKU tracks have less than  $n$  clusters
  - Number of clusters in TKD track; reject event if all TKU tracks have less than  $n$  clusters
  - TKD momentum
  - TKD chi2
  - TKU chi2
  - Exactly one space point in TOF2
  - TOF12 time
  - Upstream aperture cuts; reject event if the track radius of the global upstream track extrapolated to the virtual plane with a particular station number is more than some value. Aperture cuts should be implemented for each of the principal apertures, namely:
    - diffuser
    - TKU butterfly
    - Upstream PRY
    - lh2 upstream window
    - absorber centre
    - lh2 downstream window flange
    - SSD aperture
    - TKD He window
    - TKD stations
  - Track matching between TKU and TOF1
  - Track matching between TKU and TOF0
  - Residual position between extrapolated track from TKU to TOF1
  - Residual position between extrapolated track from TKU to TOF0
  - Residual time between extrapolated track from TKU and TOF1 to TOF0
  - Residual time between extrapolated track from TKU and TOF1 to TOF2
  - Residual position between extrapolated track from TKU to TKD and any TKD track
  - Residual momentum between extrapolated track from TKU to TKD and any TKD track
  - Residual position between extrapolated track from TKD to TOF2