Subject: [SOLVED]Eta Prime decay via rho dilepton
Posted by Michael Kunkel on Sun, 25 Aug 2013 22:14:38 GMT
View Forum Message <> Reply to Message

Greetings,

I am trying to simulate the reaction g + p -> p eta'; eta'->rho dilepton; rho->pi+ pi-.

When I do this I have to set a PInclusiveModel for the dilepton, or there is no distribution. Here is a plot without setting a PInclusiveModel.

The problem is when setting this PInclusiveModel, the invariant mass of pi+pi- is not longer rho but instead a distribution.

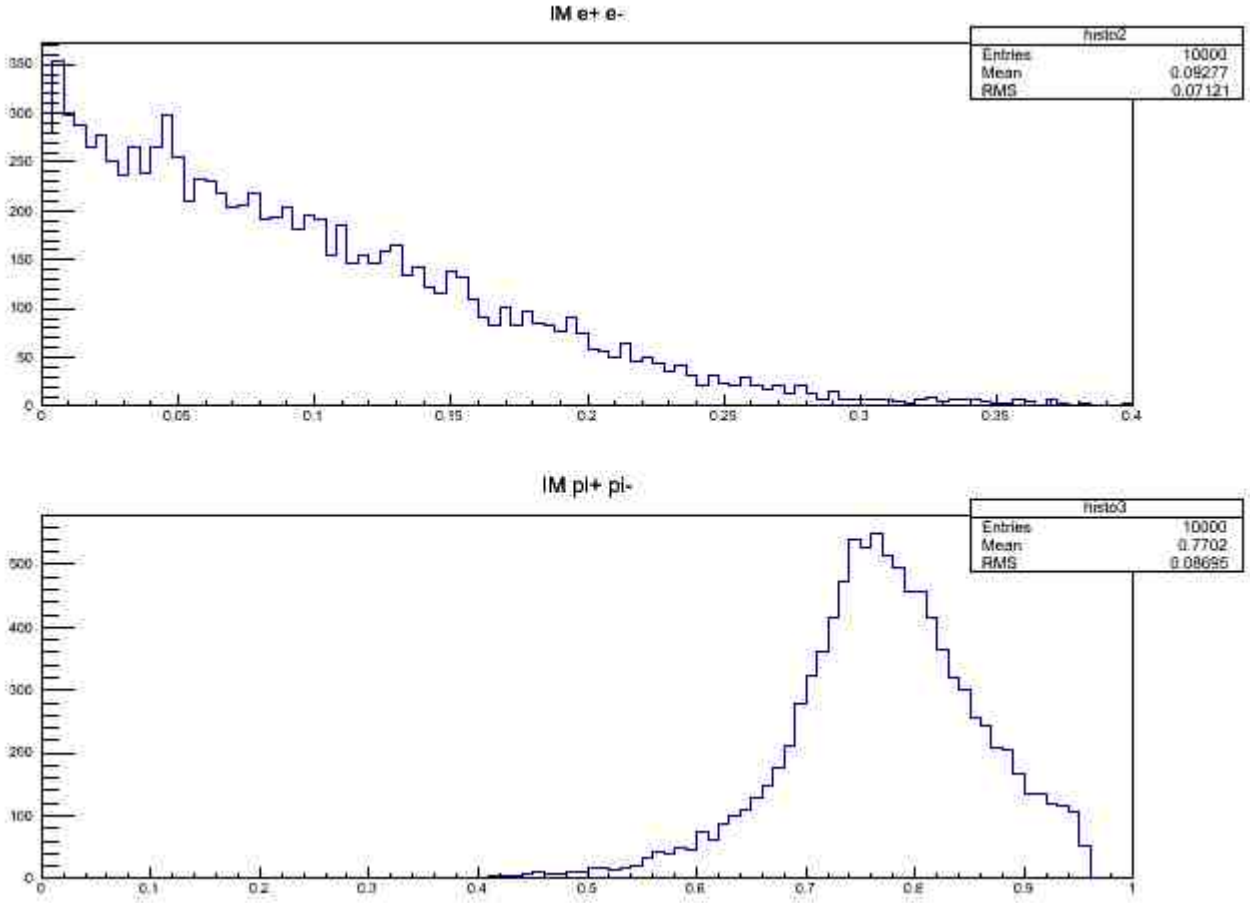
Here is a working code.

```
//#include "loadPluto.h";
//Author Michael C. Kunkel
#include "TH1.h"
#include "TH2.h"
#include "TH3.h"
#include "TChain.h"
#include "TCanvas.h"
#include "TF1.h"
#include "/Users/Mike/Pluto/pluto_v5.42/src/PParticle.h"
#include "/Users/Mike/Pluto/pluto_v5.42/src/PReaction.h"
#include "/Users/Mike/Pluto/pluto_v5.42/src/PBeamSmearing.h"
#include "/Users/Mike/Pluto/pluto_v5.42/src/PAnyDistribution.h"


void SIMULATE_EtaPrime_Dalitz(){

  makeStaticData()->AddDecay(-1, "eta' -> rho0 + dilepton ", "eta'", "rho0,dilepton",0.0009);

  //TF1 object representing the di-lepton statistics:
  TF1 *flat = new TF1("flat","1",0,1);

  //The "PInclusiveModel" can be used as a generator:
  PInclusiveModel *dilepton_generator = new
PInclusiveModel("flat@eta'_to_rho0_dilepton/generator","Dilepton generator",-1);

  //The distribution template:
  dilepton_generator->Add("eta',parent");
  dilepton_generator->Add("rho0,daughter");
  dilepton_generator->Add("dilepton,daughter,primary");
  dilepton_generator->SetSampleFunction(flat);

  //Enable distribution as a generator
```

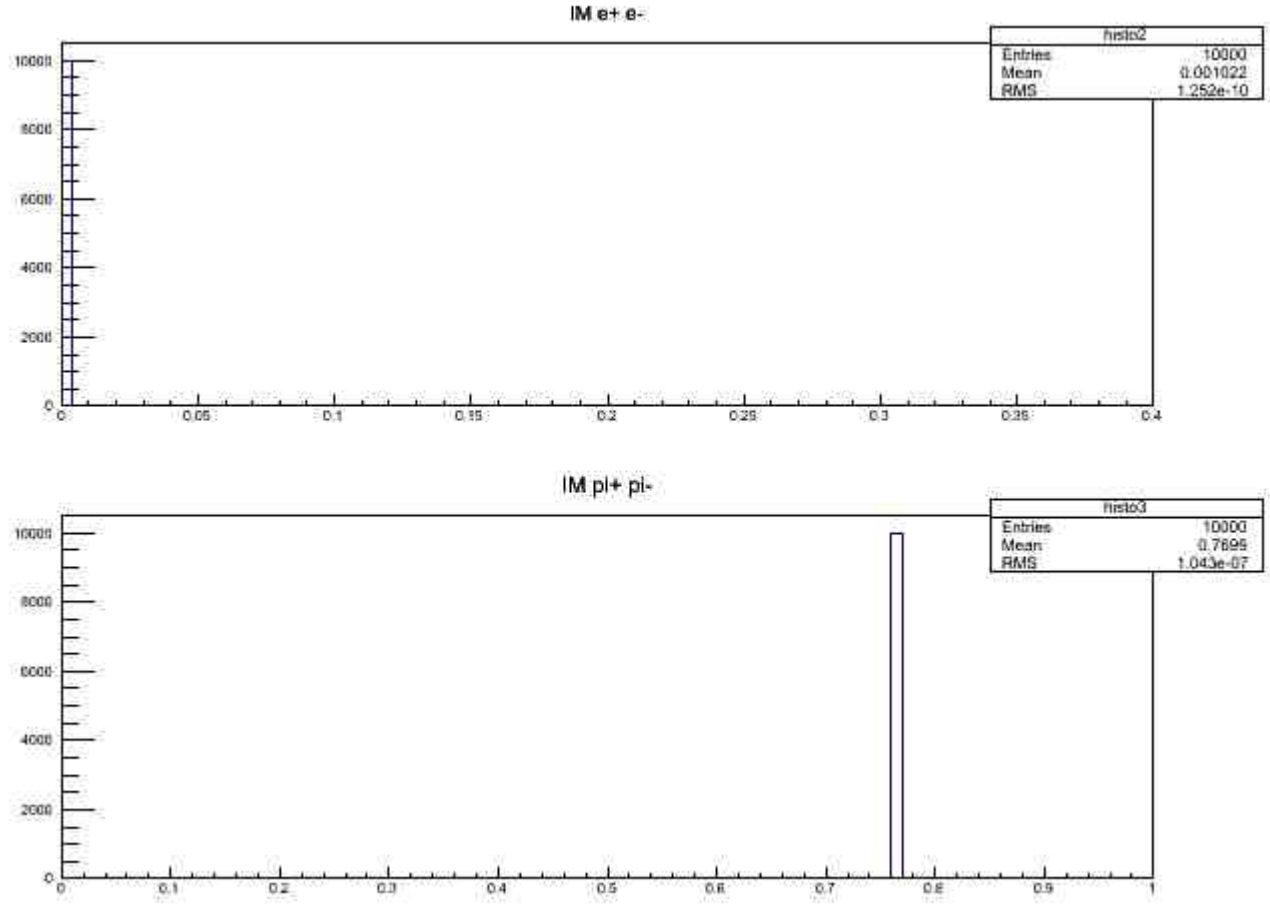
```
  //dilepton_generator->EnableGenerator();
  makeDistributionManager()->Add(dilepton_generator);

  double ebeam_min = 1.1725;
  double ebeam_max = 5.44575;
  PBeamSmearing *beam_smear = new PBeamSmearing("beam_smear", "Beam smearing");
  TF1* beam_smear_fn = new TF1("beam_smear_fn", "1./x", ebeam_min, ebeam_max);

  beam_smear->SetReaction("g + p");
  beam_smear->SetMomentumFunction(beam_smear_fn);
  makeDistributionManager()->Add(beam_smear);

  gROOT->Reset();
  //PUtils::SetSeed(123); //this is to have a fixed SEED. By default, the systime is used....

  PReaction my_reaction("_P1 = 2.2","g","p","p eta' [dilepton [e+ e-] rho0 [pi+
pi-]]","etaP_Aphi",1,0,1,1);

  TH1F * histo2 = new TH1F ("histo2","IM e+ e-",100,0.0,0.4);
  TH1F * histo3 = new TH1F ("histo3","IM pi+ pi-",100,0.0,1);

  my_reaction.Preheating(100);

  my_reaction.Do(histo2,"_w =1;_x = ([dilepton])->M() ");
  my_reaction.Do(histo3,"_w =1;_x = ([pi+] + [pi-])->M() ");
  my_reaction.Loop(10000);

  TCanvas *c1 = new TCanvas("c1","c1");
  c1->Divide(1,2);
  c1->cd(1);
  histo2->Draw();
  c1->cd(2);
  histo3->Draw();
  c1->Print("mass_spectrum.jpeg");




}
```

File Attachments

1) mass_spectrum.jpeg, downloaded 2023 times



2) mass_spectrum_nodilep_dist.jpeg, downloaded 2100 times



I worked around it by

```
//#include "loadPluto.h";
//Author Michael C. Kunkel
#include "TH1.h"
#include "TH2.h"
#include "TH3.h"
#include "TChain.h"
#include "TCanvas.h"
#include "TF1.h"
#include "/Users/Mike/Pluto/pluto_v5.42/src/PParticle.h"
#include "/Users/Mike/Pluto/pluto_v5.42/src/PReaction.h"
#include "/Users/Mike/Pluto/pluto_v5.42/src/PBeamSmearing.h"
#include "/Users/Mike/Pluto/pluto_v5.42/src/PAnyDistribution.h"


void SIMULATE_EtaPrime_Dalitz(){
 Double_t m_rho =   makeStaticData()->GetParticleMass("rho0");
  makeStaticData()->AddParticle(-1,"X", m_rho);
  makeStaticData()->AddDecay(-1,"X -> pi+ + pi=", "X", "pi+,pi-", 1.);
  //makeStaticData()->AddDecay(-1, "eta' -> rho0 + dilepton ", "eta'", "rho0,dilepton",0.0009);
  makeStaticData()->AddDecay(-1, "eta' -> X + dilepton ", "eta'", "X,dilepton",0.0009);

  //TF1 object representing the di-lepton statistics:
  TF1 *flat = new TF1("flat","1",0,1);

  //The "PInclusiveModel" can be used as a generator:
  PInclusiveModel *dilepton_generator = new
PInclusiveModel("flat@eta'_to_rho0_dilepton/generator","Dilepton generator",-1);

  //The distribution template:
  dilepton_generator->Add("eta',parent");
  dilepton_generator->Add("X,daughter");
  dilepton_generator->Add("dilepton,daughter,primary");
  dilepton_generator->SetSampleFunction(flat);

  //Enable distribution as a generator
  //dilepton_generator->EnableGenerator();
  makeDistributionManager()->Add(dilepton_generator);

  double ebeam_min = 1.1725;
  double ebeam_max = 5.44575;
  PBeamSmearing *beam_smear = new PBeamSmearing("beam_smear", "Beam smearing");
  TF1* beam_smear_fn = new TF1("beam_smear_fn", "1./x", ebeam_min, ebeam_max);

  beam_smear->SetReaction("g + p");
  beam_smear->SetMomentumFunction(beam_smear_fn);
  makeDistributionManager()->Add(beam_smear);
```

```
  gROOT->Reset();
  //PUtils::SetSeed(123); //this is to have a fixed SEED. By default, the systime is used....

  PReaction my_reaction("_P1 = 2.2","g","p","p eta' [dilepton [e+ e-] X [pi+
pi-]]","etaP_Aphi",1,0,1,1);

  TH1F * histo2 = new TH1F ("histo2","IM e+ e-",100,0.0,0.4);
  TH1F * histo3 = new TH1F ("histo3","IM pi+ pi-",100,0.0,1);

  my_reaction.Preheating(100);

  my_reaction.Do(histo2,"_w =1;_x = ([dilepton])->M() ");
  my_reaction.Do(histo3,"_w =1;_x = ([pi+] + [pi-])->M() ");
  my_reaction.Loop(10000);

  TCanvas *c1 = new TCanvas("c1","c1");
  c1->Divide(1,2);
  c1->cd(1);
  histo2->Draw();
  c1->cd(2);
  histo3->Draw();
  c1->Print("mass_spectrum_nodilep_dist.jpeg");




}
```

Subject: Re: [SOLVED]Eta Prime decay via rho dilepton
Posted by Ingo Fröhlich on Wed, 28 Aug 2013 09:49:02 GMT
View Forum Message <> Reply to Message

I don't think that we have a model for eta' -> dilepton + rho0 in hand. If you use a "generator" model for the dilepton, it does not use the M1-model for Breit-Wigner-sampling of the daughter, but a fixed mass.

PS: if you use "/generator", it does not use the PInclusiveModel as a primary one. Pluto might be confused about this. "/generator" is normally used together with a primary model in weighting mode.
