

Event Generation with EvtGen

- Current event generators focus on modeling production
 - Particle decays generally handled in Geant(4), genr8, ...
 - But Sean (you say), I want to analyze a (narrow) final state with a more complicated decay topology/distribution
- Solution: **EvtGen** library for generating sequential decays
 - Industry standard solution: BaBar, Belle(-II), CLEO(-c), CDF, D0, LHCb, ATLAS, CMS, PANDA, etc.
 - Existing library for dozens of decays
 - Website: <https://evtgen.hepforge.org/>
 - Example use cases: Multibody decays (Dalitz decays, rare decays like $\eta \rightarrow \pi^+\pi^-e^+e^-$), decay chains (e.g. $\eta' \rightarrow \pi\pi\eta$, $\eta \rightarrow 3\pi$)

Event Generation with EvtGen

- Or something more complicated:
 - $\gamma p \rightarrow D^{*0} \Lambda_c^+$
 - $D^{*0} \rightarrow D^0 \pi^0, D^0 \rightarrow K_S \pi\pi$
 - $\Lambda_c^+ \rightarrow p K \pi$
- Example stolen from documentation:

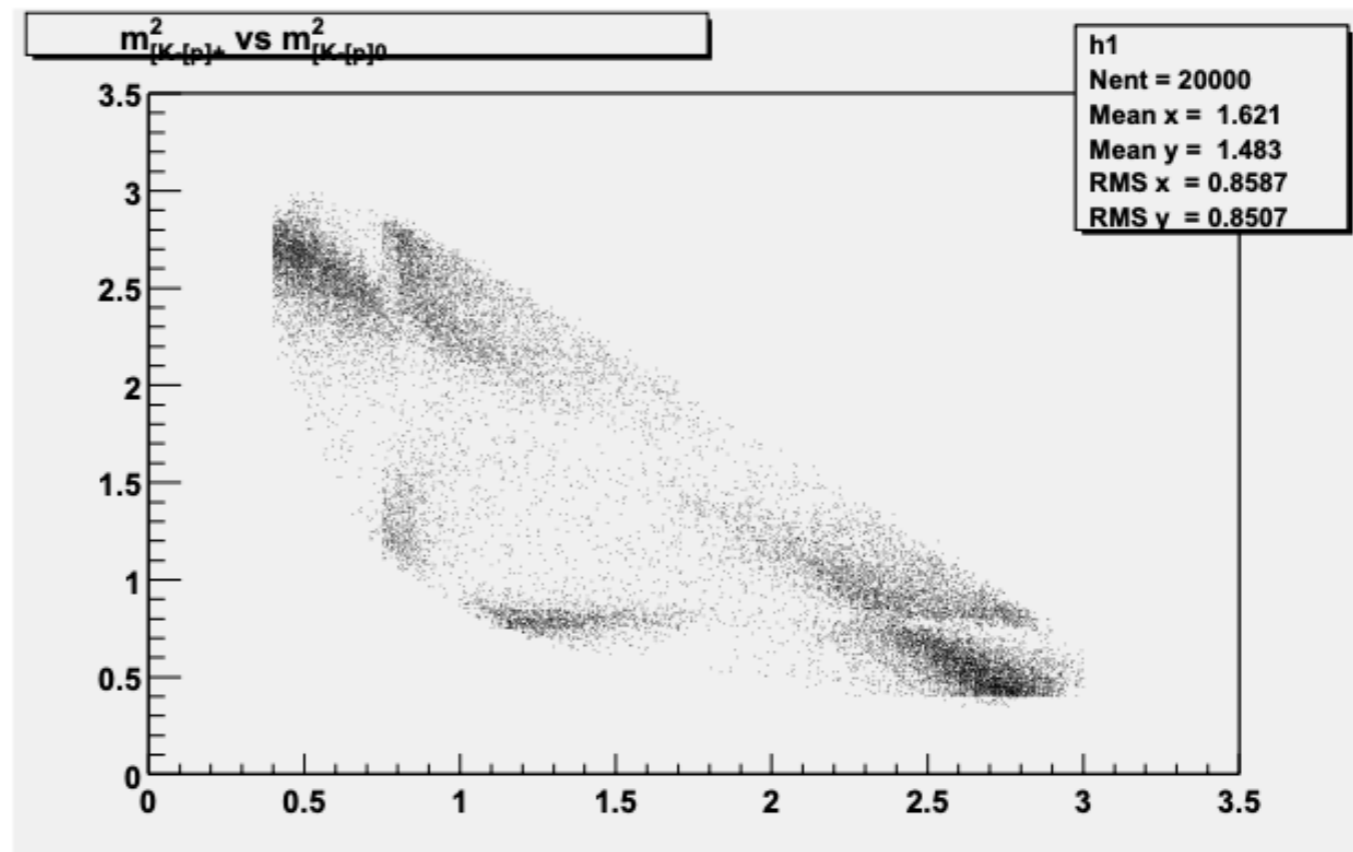


Figure 3: Histogram number 1 in ddalitz.root for 20000 event shows the dalitz plot for the $D \rightarrow K\pi\pi$ decay.

Event Generation with EvtGen: Solution 1

- Modified genEtaRegge to use EvtGen to internally decay eta's
 - Maybe I don't like Simon's decay definitions and don't want to hack the code too much
- Example decay file:

```
[sdobbs@charm genEtaRegge]$ cat eta548-evtgen.in
#beam configuration file
cobrem.conf
#mass[GeV] width[GeV]
0.547853 0.000
#g_eta_gamma_gamma g_rho_eta_gamma g_omega_eta_gamma g_phi_eta_gamma
0.0429 0.81 0.29 0.38
#Number of decay particles
-1
#EvtGen particle name
eta
```



This is new!

Event Generation with EvtGen: Solution 2

- Generated $\gamma p \rightarrow \pi^0 \eta p$ events with **gen_amp**
- Generated π^0, η decays with **decay_evtgen**
 - Reads in HDDM output from other generators, decays particles which have not already been decayed
 - Skips long-lived particles
- Default Decay file:

```
#
Decay eta
0.393100000 gamma gamma PHSP; #[Reconstructed
0.325700000 pi0 pi0 pi0 PHSP; #[Reconstructed
0.227400000 pi- pi+ pi0 ETA_DALITZ; #[Reconst
0.046000000 gamma pi- pi+ PHSP; #[Reconstructed
0.007000000 gamma e+ e- PHSP; #[Reconstructed
0.000310000 gamma mu+ mu- PHSP; #[Reconstructed
0.000270000 gamma gamma pi0 PHSP; #[Reconstructed
0.000214200 pi+ pi- e+ e- PHSP; #[Reconstructed
0.000005800 mu+ mu- PHSP; #[New mode ad
Enddecay
```

Event Generation with EvtGen: Solution 2

- Generated $\gamma p \rightarrow \pi^0 \eta p$ events with **gen_amp**
- Generated π^0, η decays with **decay_evtgen**
 - Reads in HDDM output from other generators, decays particles which have not already been decayed
 - Skips long-lived particles
 - User Decay file: (userDecay.dec)

```
Decay eta
1.000 pi-      pi+      pi0      ETA_DALITZ;
Enddecay

End
```

Must be in current working directory

Another example: omega decays

- Many different angular distributions are already built in

```
#
Decay omega
0.892000000 pi-      pi+      pi0      OMEGA_DALITZ; #[Reconstructed PDG2011]
0.082800000 pi0      gamma
0.015300000 pi-      pi+
0.000460000 eta      gamma
0.000770000 pi0      e+      e-
0.000130000 pi0      mu+     mu-
0.00150      pi+ pi- gamma      PHSP;
0.000066000 pi0      pi0      gamma      PHSP; #[Reconstructed PDG2011]
0.00050      pi+ pi- pi+ pi-      PHSP;
0.000072800 e+      e-
0.000090000 mu+     mu-
Enddecay
```

**Branching
fraction**

Decay particles

Decay model

Using EvtGen

- This is all sitting on a halld_sim branch: **sdobbs_evtgen**
 - Requires additional libraries to be built: now added to build_scripts
- Config files:
 - \$EVTGEN_HOME/DECAY.DEC — Decay definitions
[control via EVTGEN_DECAY_FILE env var]
 - \$EVTGEN_HOME/evt.pdl — Particle definitions
[control via EVTGEN_PARTICLE_DEFINITIONS]
 - userDecay.dec — Override standard decay definitions
- Feedback is welcome!