Update on $\nu\bar{\nu}\mathrm{H}, \mathrm{H} \rightarrow \mu\mu$

C. Calancha
calancha@post.kek.jp

March 14, 2014

Momentum Resolution Event-by-Event Basis

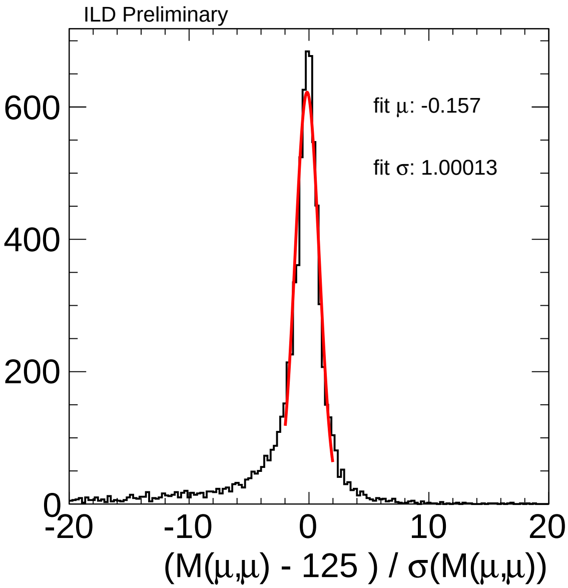
- Working on my code during last week.
- More efficient code
- Remove unnecessary intermediate steps.
- Improve readability of the code.
- I am using it for muons comming from $\mathrm{H} \rightarrow \mu\mu$ but can be used with whatever charged particle.

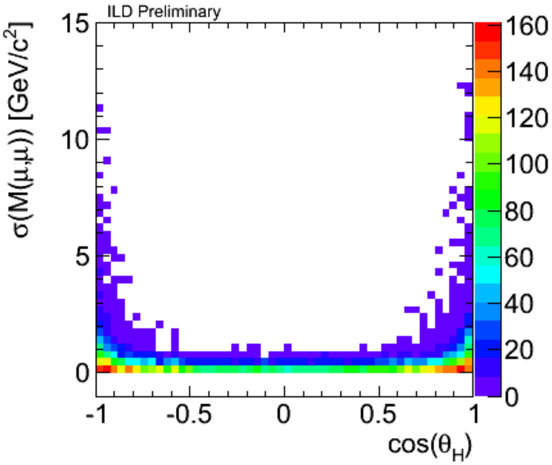
Code

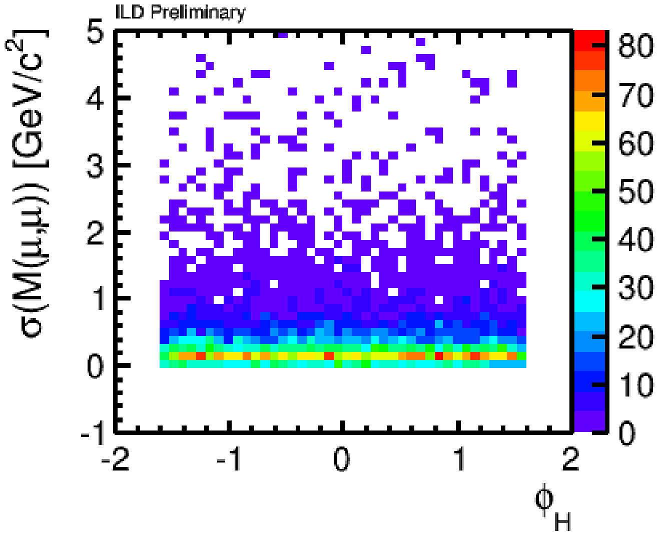
The code is available here: http://www-jlc.kek.jp/jlc/en/node/202

- Current version of the code uses momenta from PandoraPFO collection.
- If you perform some kind of FSAR recovery, those particle momenta will change.
  - You should use the corrected momenta in the dimuon mass error calculation (current code does not).
  - I would expect this error would decrease (if you are really doing a correction).
    $\longrightarrow$ Could be useful to test our FSAR correction algorithms.

Gaussian Fit [-2,2]




ILD Preliminary
σ(M(μ,μ)) [GeV/c²]
cos(θ_H)
15
10
5
0
-1
-0.5
0
0.5
1
160
140
120
100
80
60
40
20
0


ILD Preliminary
σ(M(μ,μ)) [GeV/c²]
5
4
3
2
1
0
-1
-2
-1
0
1
2
ϕ_H
80
70
60
50
40
30
20
10
0

Status / Plan

Summary

- Developed piece of code to get covariance matrix of charged particle in momenta space.
- Used it to calculate dimuon mass error event-by-event.
    - Dimuon mass error variable looks sensible.

Plan

- Proceed with $\mathrm{H} \rightarrow \mu\mu$ updated analysis.

Back Up

Back Up

$\mathrm{H} \rightarrow \mu\mu$

Update on $\mathrm{H} \rightarrow \mu\mu$ with improved analysis methods

- Started to work on $\mathrm{H} \rightarrow \mu\mu$ update.

Plan

1. Final selection with TMVA.
2. Weigth on $\sigma(\mathrm{M}(\mu^{+}, \mu^{-}))$ event by event (Mikael suggestion).

Weight on $\sigma(\mathrm{M}(\mu^+, \mu^-))$

Mikael Suggestion

- Dimuon mass given by muon momentum measurement.
  - Variance on $\mathrm{M}(\mu^+, \mu^-)$ $\longrightarrow$ error propagation from momentum uncertainty.
- Well measured signal events have small $\mathrm{M}(\mu^+, \mu^-)$ errors.
- Badly measured background events lying around signal window would have larger $\mathrm{M}(\mu^+, \mu^-)$ errors.

$\longrightarrow$ Variable $1/\sigma(\mathrm{M}(\mu^+, \mu^-))$ could improve S/B.

($\mathrm{M}(\mu^+, \mu^-)$ depends simetrically on both muons)

$$\sigma^2(\mathrm{M}(\mu^+, \mu^-)) = \frac{1}{\mathrm{M}(\mu^+, \mu^-)^2}[\mathrm{P}_1^{\mathrm{T}}\,\Sigma_2\,\mathrm{P}_1 + \mathrm{P}_2^{\mathrm{T}}\,\Sigma_1\,\mathrm{P}_2]$$

(*) P_i column vector $x^i = (E, px, py, pz)$, P_i^T, row vector $x_i = g_{ij}x^j = (E, -px, -py, -pz)$

(g_{ij} metric in the Minkowski space: [+1 -1 -1 -1]), Σ_i covariance matrix of muon i.

Weight on $\sigma(\mathrm{M}(\mu^+, \mu^-))$

$$\sigma^2(\mathrm{M}(\mu^+, \mu^-)) = \frac{1}{\mathrm{M}(\mu^+, \mu^-)^2}[\mathrm{P}_1^{\mathrm{T}}\, \Sigma_2\, \mathrm{P}_1 + \mathrm{P}_2^{\mathrm{T}}\, \Sigma_1\, \mathrm{P}_2]$$

(*) P_i column vector $x^i = (E, px, py, pz)$, P_i^T, row vector $x_i = g_{ij} x^j = (E, -px, -py, -pz)$ (g_{ij} metric in the Minkowski space: [+1 -1 -1 -1]), Σ_i covariance matrix of muon i.

- I assumed $Var(m_i) = 0$ (m_i, muon mass).
- Even without mass constraint, at 1 TeV, i would expect lim $\frac{Var(m)}{Var(p)} \approx 0$

ReconstructedParticle.getCovMatrix() returns null

- This covariance matrix is not filled (why not?).

$\longrightarrow$ Need to perform additional algebra.

  - Covariance matrix from associated track is available.
  - In helix parameterization basis: $d_0, z_0, \Omega, \phi, tan\lambda$
  - If $\mathfrak{J}$ is the Jacobian matrix from helix parameters to (px,py,pz,E)

$\longrightarrow$ $\Sigma_i' = \mathfrak{J}\, \Sigma_i\, \mathfrak{J}^T$, covariance matrix in momenta space.

Jacobian helix parameters to momenta space

- After some derivative exercises ...

$$\mathfrak{J} = \begin{bmatrix} \frac{\partial P_x}{\partial tan\lambda} & \frac{\partial P_y}{\partial tan\lambda} & \frac{\partial P_z}{\partial tan\lambda} & \frac{\partial E}{\partial tan\lambda} \\ \frac{\partial P_x}{\partial \Omega} & \frac{\partial P_y}{\partial \Omega} & \frac{\partial P_z}{\partial \Omega} & \frac{\partial E}{\partial \Omega} \\ \frac{\partial P_x}{\partial d_0} & \frac{\partial P_y}{\partial d_0} & \frac{\partial P_z}{\partial d_0} & \frac{\partial E}{\partial d_0} \\ \frac{\partial P_x}{\partial z_0} & \frac{\partial P_y}{\partial z_0} & \frac{\partial P_z}{\partial z_0} & \frac{\partial E}{\partial z_0} \\ \frac{\partial P_x}{\partial \phi} & \frac{\partial P_y}{\partial \phi} & \frac{\partial P_z}{\partial \phi} & \frac{\partial E}{\partial \phi} \end{bmatrix} = \frac{-1}{\Omega} \begin{bmatrix} 0 & 0 & -\Omega P_T & -\frac{P_z^2\,\Omega}{E\,tan\lambda} \\ P_x & P_y & P_z & \frac{P^2}{E} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ P_y\,\Omega & -P_x\,\Omega & 0 & 0 \end{bmatrix}$$

$\longrightarrow$ $\Sigma_i' = \mathfrak{J}\,\Sigma_i\,\mathfrak{J}^T$, covariance matrix in momenta space.

- Could be useful storing this covariance matrix for charged particles.
- so that, we could retrieve it by calling ReconstructedParticle.getCovMatrix(). (Currently returning null matrix)