Contents

GamPola

Author:
Borys KNYSH

1. Theoretical framework

$B^+ \to K^+\pi^+\pi^-\gamma$
$B^0 \to K^+\pi^0\pi^-\gamma$
This section contains overview of theoretical expressions used in order to describe differential decay rate.

1.1. *Kinematics of* $B \to K\pi\pi\gamma$ *decay*

This subsection contains the expressions of the kinematical variables used to describe $B \to K\pi\pi\gamma$ decay. Majority of expressions are incapsulated in the *GKinematics* module, but some of them such as Lorenz's boosts and Euler's rotations are included in the modules *EvtGVector3D* and *EvtGVector4D*. Following are given building blocks which depend on each other:

- Defining Lorenz's transformation:

$$E = \gamma(E' + (\overrightarrow{\beta} \cdot \overrightarrow{p}')) \quad (1)$$

$$\overrightarrow{p} = \overrightarrow{p}' + (\gamma - 1)\overrightarrow{\beta}\frac{(\overrightarrow{\beta} \cdot \overrightarrow{p}')}{\beta^2} + \gamma\overrightarrow{\beta}E' \quad (2)$$

- Defining Euler's rotations:

$$v_x = v_{0x}(\cos\chi\cos\theta\cos\varphi - \sin\chi\sin\varphi) + v_{0y}(-\sin\chi\cos\theta\cos\varphi - \cos\chi\sin\varphi) + v_{0z}\sin\theta\cos\varphi \quad (3)$$

$$v_y = v_{0x}(\cos\chi\cos\theta\sin\varphi + \sin\chi\cos\varphi) + v_{0y}(-\sin\chi\cos\theta\sin\varphi + \cos\chi\cos\varphi) + v_{0z}\sin\theta\sin\varphi \quad (4)$$

$$v_z = v_{0x}(-\cos\chi\sin\theta) + v_{0y}(\sin\chi\sin\theta) + v_{0z}\cos\theta \quad (5)$$

- Calculating energy and momentum of final state particles in the rest frame of decaying particle in 2-body decay (*GKinematics::GMomentaA*, *GKinematics::GEnergyA*):

$$p(s_{ijk}, s_{ij}, m_b) = \sqrt{\frac{(s_{ijk} - (\sqrt{s_{ij}} + m_b)^2))}{4s_{ijk}}} \quad (6)$$

$$E(s_{ijk}, s_{ij}, m_b) = \frac{s_{ijk} - m_b^2 + s_{ij}}{2\sqrt{s}_{ijk}} \quad (7)$$

- Initialization the following kinematical constants using Eq. 7: $E_{K^*_{892}}, E_{\rho_{775}}, E_{K^*_{1430}}$. They will be required for helicity amplitudes calculation;

- Calculating lower and upper limits of the Dalitz plot (*GKinematics::GSjkMin* and *GKinematics::GSjkMax*):

$$E_j^* = E(s_{ij}, m_i^2, m_j) \quad (8)$$

$$E_k^* = \frac{s_{ijk} - s_{ij} - m_k^2}{2\sqrt{s_{ij}}} \quad (9)$$

$$s_{jk}^{low} = (E_j^* + E_k^*)^2 - (\sqrt{E_j^{2*} - m_j^2} + \sqrt{E_k^{2*} - m_k^2})^2 \quad (10)$$

$$s_{jk}^{up} = (E_j^* + E_k^*)^2 - (\sqrt{E_j^{2*} - m_j^2} - \sqrt{E_k^{2*} - m_k^2})^2 \quad (11)$$

- Defining the dot product of two pions' 4 momentum (*GKinematics::GPi1Pi2Vec4*), which will be used for calculating 1^+ ampiltudes;

$$p_i \cdot p_j = \frac{s_{K\pi_1\pi_2} - (s_{K\pi_1} + s_{K\pi_2}) + m_K^2}{2} \quad (12)$$

- Defining the cross product of two pions' 3 momentum (*GKinematics::GPiPjVec3*). Note that this cross product is used to define 2^+ amplitude;

- Calculating dependent dalitz variable (*GKinematics::GSij*), which will be usefull for calculation of $\varphi_K, \varphi_{\pi_1}, \varphi_{\pi_2}$ angles:

$$s_{ik} = s_{ijk} - s_{ij} - s_{jk} + m_i^2 + m_j^2 + m_k^2 \quad (13)$$

- Calculating dot product of two pions' 3D vectors (*GKinematics::GPiPj*). Such product is used for calculating $\varphi_K, \varphi_{\pi_1}, \varphi_{\pi_2}$ angles, which will be required for 4-momentum calculation of K, π_1, π_2 in the K_{res}-rest frame:

$$(\overrightarrow{p}_{\pi_1} \cdot \overrightarrow{p}_{\pi_2}) = \frac{E_{\pi_1}E_{\pi_2} + (m_{\pi_1}^2 + m_{\pi_2}^2 - s_{\pi_1\pi_2})}{2} \tag{14}$$

- Calculating scalar product of two arbitrary 3D-vectors:

$$(\overrightarrow{v}_i \cdot \overrightarrow{v}_j) = v_{i1}v_{j1} + v_{i2}v_{j2} + v_{i3}v_{j3} \tag{15}$$

  Required to define 2^+ amplitude;

- Calculating $[\overrightarrow{p}_{\pi_1} \times \overrightarrow{p}_{\pi_2}]$. Will be used for calculating 2^+ amplitude (*GKinematics::GPi1Pi2Vec3*):

$$[\overrightarrow{p}_{\pi_1} \times \overrightarrow{p}_{\pi_2}] = (p_{\pi_1 y}p_{\pi_2 z} - p_{\pi_1 z}p_{\pi_2 y})\overrightarrow{i} + (p_{\pi_1 z}p_{\pi_2 x} - p_{\pi_1 x}p_{\pi_2 z})\overrightarrow{j} + (p_{\pi_1 x}p_{\pi_2 y} - p_{\pi_2 x}p_{\pi_1 y})\overrightarrow{k} \tag{16}$$

- Generation of valid phase space of three body decay (*GKinematics::GGetPhaseSpace*):

Algorithm 1: Phase space generation

Result: $(\cos\theta, \varphi, s_{K\pi_1}, s_{\pi_1\pi_2}, s_{K\pi\pi})$
while *1* **do**
  $s_{K\pi\pi} = uniform(s_{min}, s_{max})$;
  $s_{\pi\pi} = uniform((m_{1\pi} + m_{2\pi})^2, (\sqrt{s_{K\pi\pi}} - m_K)^2)$;
  $s_{K\pi} = uniform((m_K + m_\pi)^2, (\sqrt{s_{K\pi\pi}} - m_\pi)^2)$;
  $s_{\pi\pi}^{min}$ and $s_{\pi\pi}^{max}$ calculations using Eqs. 10-11;
  if $s_{\pi\pi} \geq s_{\pi\pi}^{min}$ *and* $s_{\pi\pi} \leq s_{\pi\pi}^{max}$ **then**
    break the loop;
  end
end
$\cos\theta = uniform(-1, 1)$;
$\varphi = uniform(0, 2\pi)$;

- Following expressions are used to calculate all the kinematical variables used for helicity amplitudes calculation (*GKinematics::GCalculateKinematics*). These computations are performed for each event:
  - Defining the $\cos\tilde{\theta}$ which is needed for neutral decay modes (without it, the integration of the Dalitz variables cancel the A_{ud}-asymmetry)

$$\cos\tilde{\theta} = \begin{cases} \cos\theta & charge = 1 \\ \cos\theta & charge = 0, s_{K\pi_1} \geq s_{K\pi_2} \\ -\cos\theta & charge = 0, s_{K\pi_1} < s_{K\pi_2} \end{cases}, \tag{17}$$

    where $\cos\theta$ is provided[1];
  - Calculating momentum and energies of pion and kaon using Eqs.6-7;
  - Calculating energy and momentum of photon in K_{res} rest frame:

$$E_\gamma = |\overrightarrow{p_\gamma}| = \frac{m_B - s_{K\pi\pi}}{2\sqrt{s_{K\pi\pi}}}; \tag{18}$$

  - Calculating scalar product of two 3 and 4-dimensional momenta of pions using Eqs.12-14;
  - Defining angle between two pions in the K_{res} rest frame using Eqs. 6,14:

$$\delta = \arccos\left(\frac{(\overrightarrow{p}_{\pi_1} \cdot \overrightarrow{p}_{\pi_2})}{|\overrightarrow{p}_{\pi_1}| \cdot |\overrightarrow{p}_{\pi_2}|}\right) \tag{19}$$

  - Defining angle between K and π_1 using Eqs. 6,19:

$$\alpha = \arctan\left(\frac{|\overrightarrow{p}_{\pi_2}| \sin\delta}{|\overrightarrow{p}_{\pi_1}| + |\overrightarrow{p}_{\pi_2}| \cos\delta}\right) \tag{20}$$

[1] Despite the fact that helicity amplitudes are calculated using modified $\cos\tilde{\theta}$, distribution of $\cos\theta$ is sampled from the p.d.f.

– Defining azimuthal angles between x-axis and corresponding final state particles in the $K\pi\pi$ rest-frame:

$$\varphi_{\pi_1} = \frac{2\varphi - \delta}{2} \quad (21)$$

$$\varphi_{\pi_2} = \frac{2\varphi + \delta}{2} \quad (22)$$

$$\varphi_K = \pi + \alpha + \varphi - \frac{\delta}{2} \quad (23)$$

– The pion's modified 3-momenta in the $\tilde{\theta}, \phi_{\pi,K}$ cordinates (used for calcualtion helicity currents):

$$\overrightarrow{p}_{\pi_1} = |\overrightarrow{p}_{\pi_1}|(\cos\tilde{\theta}\cos\varphi_{\pi_1}; \sin\varphi_{\pi_1}; -\sin\tilde{\theta}\cos\varphi_{\pi_1}) \quad (24)$$

$$\overrightarrow{p}_{\pi_2} = |\overrightarrow{p}_{\pi_2}|(\cos\tilde{\theta}\cos\varphi_{\pi_2}; \sin\varphi_{\pi_2}; -\sin\tilde{\theta}\cos\varphi_{\pi_2}) \quad (25)$$

$$\overrightarrow{p}_K = |\overrightarrow{p}_K|(\cos\tilde{\theta}\cos\varphi_K; \sin\varphi_K; -\sin\tilde{\theta}\cos\varphi_K) \quad (26)$$

– Calculating $([\overrightarrow{\tilde{p}_{\pi_1}} \times \overrightarrow{\tilde{p}_{\pi_2}}] \cdot \overrightarrow{\varepsilon_0})$, $(\overrightarrow{p}_{\pi_1} \cdot \overrightarrow{\varepsilon_0})$ and $(\overrightarrow{p}_{\pi_2} \cdot \overrightarrow{\varepsilon_0})$ using Eqs.15-16;

– Calculating kinematical coefficients for 2^+ amplitude:

$$\overrightarrow{K}_{C4} = (\overrightarrow{p}_{\pi_1} \cdot \overrightarrow{\varepsilon_0}) \cdot [\overrightarrow{p}_{\pi_1} \times \overrightarrow{p}_{\pi_2}] + ([\overrightarrow{p}_{\pi_1} \times \overrightarrow{p}_{\pi_2}] \cdot \overrightarrow{\varepsilon_0}) \cdot \overrightarrow{p}_{\pi_1} \quad (27)$$

$$\overrightarrow{K}_{C5} = (\overrightarrow{p}_{\pi_2} \cdot \overrightarrow{\varepsilon_0}) \cdot [\overrightarrow{p}_{\pi_1} \times \overrightarrow{p}_{\pi_2}] + ([\overrightarrow{p}_{\pi_1} \times \overrightarrow{p}_{\pi_2}] \cdot \overrightarrow{\varepsilon_0}) \cdot \overrightarrow{p}_{\pi_2} \quad (28)$$

- Conversion of the generated phase space to the set of four 4-momenta of final state particles in the B rest frame:

– Defining the pion's and kaon's modified 3-momenta in the $\theta, \phi_{\pi,K}$ cordinates:

$$\overrightarrow{p}_{\pi_1} = |\overrightarrow{p}_{\pi_1}|(\cos\theta\cos\varphi_{\pi_1}; \sin\varphi_{\pi_1}; -\sin\theta\cos\varphi_{\pi_1}) \quad (29)$$

$$\overrightarrow{p}_{\pi_2} = |\overrightarrow{p}_{\pi_2}|(\cos\theta\cos\varphi_{\pi_2}; \sin\varphi_{\pi_2}; -\sin\theta\cos\varphi_{\pi_2}) \quad (30)$$

$$\overrightarrow{p}_K = |\overrightarrow{p}_K|(\cos\theta\cos\varphi_K; \sin\varphi_K; -\sin\theta\cos\varphi_K) \quad (31)$$

– Defining 4-momenta of final state particles in the K_{res} rest frame:

$$p_\gamma = \Big(E_\gamma; 0; 0; -|\overrightarrow{p}_\gamma|\Big); \quad (32)$$

$$p_{\pi_1} = \Big(E_{\pi_1}; \overrightarrow{p}_{\pi_1}); \quad (33)$$

$$p_{\pi_2} = \Big(E_{\pi_2}; \overrightarrow{p}_{\pi_2}); \quad (34)$$

$$p_K = \Big(E_K; \overrightarrow{p}_K) \quad (35)$$

– Defining boost vector from K_{res} rest frame to B rest frame:

$$s_{K\pi_1\pi_2} = E_K + E_{\pi_1} + E_{\pi_2} \quad (36)$$

$$\overrightarrow{\beta} = (0; 0; \frac{M_B^2 - s_{K\pi_1\pi_2}}{M_B^2 + s_{K\pi_1\pi_2}}) \quad (37)$$

This generic representation allows flexible operating of well-known objects such as Lorenz vectors. In particular to have such conversion function will be extremely useful while plugging "GamPola" model to the *EvtGen*.

– Performing the boost and indroducing new 4-momenta of final state particles in the B rest frame using Eq.;

– Synchonous randomizing of the directions of final state particles in the B rest frame using Euler's angles and Eq.:

$$\alpha^{rand} = 2\pi \cdot uniform(0,1) \quad (38)$$

$$\beta^{rand} = \arccos(uniform(-1,1)) \quad (39)$$

- Final state 4-momenta to phase space conversion (*GKinematics::G4VectorsToPhaseSpace*) is applied when given input final state particles with their 4-momenta in the B-rest frame:

– Defining 4-momentum of K_{res}:

$$p_{K_{res}} = p_K + p_{\pi_1} + p_{\pi_2} \quad (40)$$

– Calculating $s_{K\pi_1}, s_{\pi_1\pi_2}, s_{K\pi_1\pi_2}$ (Dalitz components of phase space):

$$\begin{aligned} s_{K\pi_1} &= (p_K + p_{\pi_1})^2 && (41)\\ s_{\pi_1\pi_2} &= (p_{\pi_1} + p_{\pi_2})^2 && (42)\\ s_{K\pi_1\pi_2} &= p^2_{K_{res}} && (43) \end{aligned}$$

– Defining boosting vector from B rest frame to K_{res} rest frame:

$$\overrightarrow{\beta}_{B\to K_{res}} = -\frac{\overrightarrow{p}_{K_{res}}}{E_{K_{res}}} \tag{44}$$

– Performing boost of final state particles from B rest frame to K_{res} rest frame (will be marked as $*$) using Eq.;

– Defining normal vector to $K\pi\pi$ plane:

$$\overrightarrow{n} = \frac{[\overrightarrow{p}_{\pi 1} \times \overrightarrow{p}_{\pi_2}]}{|[\overrightarrow{p}_{\pi 1} \times \overrightarrow{p}_{\pi_2}]|} \tag{45}$$

– Defining the basis of the frame, where $\overrightarrow{e}_y$ vector is in the $K\pi\pi$ plane and $\overrightarrow{e}_x$ is in the same plane with $\overrightarrow{e}_z$ and $\overrightarrow{n}$:

$$\begin{aligned} \overrightarrow{e}_z &= -\frac{\overrightarrow{p}^*_\gamma}{E^*_\gamma} && (46)\\ \overrightarrow{e}_y &= \frac{[\overrightarrow{e}_z \times \overrightarrow{n}]}{|[\overrightarrow{e}_z \times \overrightarrow{n}]|} && (47)\\ \overrightarrow{e}_x &= [\overrightarrow{e}_y \times \overrightarrow{e}_z] && (48) \end{aligned}$$

– Calculating the angle between photon and normal to the $K\pi\pi$ plane (this angle is one of the dimensions of the phase space):

$$\cos\theta = \overrightarrow{e}_z \cdot \overrightarrow{n} \tag{49}$$

– Defining $\overrightarrow{e}'_x$ which lies in the $K\pi\pi$ plane:

$$\overrightarrow{e}'_x = \overrightarrow{e}_x \cos\theta - \overrightarrow{e}_z \sin\theta \tag{50}$$

– Since $\overrightarrow{e}'_x$ and $\overrightarrow{e}_y$ are in the same $K\pi\pi$ plane it is possible to calculate angles of two pions[2]:

$$\begin{aligned} \varphi_{\pi_1} &= \arctan\frac{(\overrightarrow{e}_y \cdot \overrightarrow{p}_{\pi_1})}{(\overrightarrow{e}'_x \cdot \overrightarrow{p}_{\pi_1})} && (51)\\ \varphi_{\pi_2} &= \arctan\frac{(\overrightarrow{e}_y \cdot \overrightarrow{p}_{\pi_2})}{(\overrightarrow{e}'_x \cdot \overrightarrow{p}_{\pi_2})} && (52) \end{aligned}$$

Hence $\varphi_{\pi 1}, \varphi_{\pi_2} \in [-\pi; \pi]$;

– Calculating last but not least component of phase space by definition:

$$\phi = \frac{\varphi_{\pi_1} + \varphi_{\pi_2}}{2} \tag{53}$$

This variable stands for combined angle of two pions in the $K\pi\pi$ plane. The value of $\phi \in [-2\pi; 2\pi]$, however originally generator produces corresponding values in the range $[0; 2\pi]$. Hence, additional shifts to the conventional region are required;

– Calculating combined angle of two pions in the valid region using Eq. 19:

$$\varphi = \begin{cases} \phi + 2\pi & \delta = \varphi_{\pi_2} - \varphi_{\pi_1}, \phi < 0 \\ \phi & \delta = \varphi_{\pi_2} - \varphi_{\pi_1}, \phi \geq 0 \\ \phi + \pi & \delta \neq \varphi_{\pi_2} - \varphi_{\pi_1} \end{cases}, \tag{54}$$

[2]in C++ arctan function is represented as atan2(y,x) function, which returns values in the range $[-\pi; \pi]$

1.2. Resonances parametrization

This subsection describes the parametrization of each resonance participating in the decay rate expression. Given below expressions are implemented within *GBreitWigner* module. Despite the fact that resonances incorporated in the currrent module are not isolated[3] this parametrization brings simplicity to the model. List of implemented resonances, their masses and widths is given in the Table.

Resonance	nominal mass, GeV	nominal width, GeV
K^{*892}	0.896	0.047
ρ^{775}	0.775	0.149
K_1^{1270}	1.272	0.090
K_1^{1400}	1.403	0.174
K^{*1410}	1.414	0.232
K^{*1680}	1.717	0.322
K_2^{1430}	1.426	0.099

Table 1. Masses and widths of implemented resonances

- Individual K_{res} are parametrized as relativistic Breit-Wigner functions with fixed width:

$$BW_{K_{res}}(s_{K\pi\pi}) = \frac{1}{s_{K\pi\pi} - M^2_{K_{res}} + i\Gamma_{K_{res}} M_{K_{res}}} \tag{55}$$

  where $M_{K_{res}}$ and $\Gamma_{K_{res}}$ stand for nominal mass and width of K_{res} resonance respectively;

- K^{*892} resonance is parametrized in the same way:

$$BW_{K^{*892}}(s_{K\pi}) = \frac{1}{s_{K\pi} - M^2_{K^{*892}} + i\Gamma_{K^{*892}} M_{K^{*892}}} \tag{56}$$

  where M_{K^*} and Γ_{K^*} stand for nominal mass and width of K^* respectively;

- ρ^{775} shape accounts for energy dependent width:

$$\Gamma_{\rho^{775}}(s_{\pi_1\pi_2}) = \Gamma_{\rho^{775}} \Big[\frac{s_{\pi_1\pi_2} - (m_{\pi_1} + m_{\pi_2})^2}{M_{\rho^{775}} - (m_{\pi_1} + m_{\pi_2})^2} \Big]^{3/2} \tag{57}$$

$$BW_{\rho^{775}} = \frac{1}{s_{\pi\pi} + M^2_{\rho^{775}} + i\Gamma_{\rho^{775}}(s_{\pi_1\pi_2}) M_{\rho^{775}}} \tag{58}$$

  where M_ρ and Γ_ρ stand for nominal mass and width of ρ respectively;

1.3. Model of $1^+ \to (1^- \to 0^- 0^-)0^-$ decays

1.3.1. *Model of quark pairs creation*

This subsection contains expressions of definition of spacial integrals of decays $A \to B + C$, S, D waves and decay amplitudes $A^{K_{res}}_{S/D}$, which were originally represented in the PhD thesis of A.Tayduganov:

- Author derives spacial integrals from harmonic oscillator meson wave functions and using Eq. 6:

$$p_{res} = p(s_{K\pi\pi}, s_{ij}, m_k^2) \tag{59}$$

$$I_0^{(ABC)} = -\frac{4\sqrt{3}}{\pi^{5/4}} \frac{R_A^{5/2}(R_B R_C)^{3/2}}{(R_A^2 + R_B^2 + R_C^2)^{5/2}} \Big(1 - p^2_{res} \frac{(2R_A^2 + R_B^2 + R_C^2)(R_B^2 + R_C^2)}{4(R_A^2 + R_B^2 + R_C^2)}\Big)$$
$$\times \exp\Big[- p^2_{res} \frac{R_A^2(R_B^2 + R_C^2)}{8(R_A^2 + R_B^2 + R_C^2)}\Big] \tag{60}$$

$$I_1^{(ABC)} = \frac{4\sqrt{3}}{\pi^{5/4}} \frac{R_A^{5/2}(R_B R_C)^{3/2}}{(R_A^2 + R_B^2 + R_C^2)^{5/2}} \exp\Big[- p^2_{res} \frac{R_A^2(R_B^2 + R_C^2)}{8(R_A^2 + R_B^2 + R_C^2)}\Big] \tag{61}$$

  where $R_A = R_B = R_C = 0.4$ — meson function raduis;

[3] Usually case Breit-Wigner parametrization is chosen for isolated resonance.

- Using $I_0^{(ABC)}, I_1^{(ABC)}$, S and D waves expressions are calculated using Eqs. 6,60,61

$$p_{res}^{const} = p(M_{K_{res}}^2, m_{ij}^2, m_k) \quad (62)$$

$$S^{(ABC)} = \gamma\sqrt{\frac{3}{2}}\frac{2I_1^{(ABC)} - I_0^{(ABC)}}{18}\exp[((p_{res}^{const})^2 - p_{res}^2)f_2] \quad (63)$$

$$D^{(ABC)} = \gamma\sqrt{\frac{3}{2}}\frac{I_1^{(ABC)} + I_0^{(ABC)}}{18}\exp[((p_{res}^{const})^2 - p_{res}^2)f_2] \quad (64)$$

where γ, f_2 — model parameters defined in the Table. $M_{K_{res}}$ — nominal mass of resonance, m_{ij}, m_k — nominal masses of pseudo vector and pseudo scalar particles obtained in the 2 body decay: $K_{res} \to PV + PS$. Exponential term is introduced in order to account for the cut-off in the coupling vertices.

- Calculating decay amplitudes of K_1 into $K^*\pi$ or $K\rho$ final states:

$$EC_{(K\pi_i)\pi_j} = 8\sqrt{s\pi^3 E(s, s_{K\pi_i}, m_{\pi_j})E(s, m_{\pi_j}^2, \sqrt{s_{K\pi_i}})} \quad (65)$$

$$EC_{K(\pi_i\pi_j)} = 8\sqrt{s\pi^3 E(s, s_{\pi_i\pi_j}, m_K)E(s, m_K^2, \sqrt{s_{\pi_i\pi_j}})} \quad (66)$$

$$\tilde{A}_S^{K_1(1270)\to(K\pi_i)\pi_j} = EC_{(K\pi_i)\pi_j} g_{1270,S}^{K\pi_i} S^{(K\pi_i\pi_j)}(\sqrt{2}\sin\theta_{K_1} - \cos\theta_{K_1}) \quad (67)$$

$$\tilde{A}_D^{K_1(1270)\to(K\pi_i)\pi_j} = EC_{(K\pi_i)\pi_j} g_{1270,D}^{K\pi_i} D^{(K\pi_i\pi_j)}(-\sin\theta_{K_1} - \sqrt{2}\cos\theta_{K_1}) \quad (68)$$

$$\tilde{A}_S^{K_1(1400)\to(K\pi_i)\pi_j} = EC_{(K\pi_i)\pi_j} g_{1400,S}^{K\pi_i} S^{(K\pi_i\pi_j)}(\sqrt{2}\cos\theta_{K_1} + \sin\theta_{K_1}) \quad (69)$$

$$\tilde{A}_D^{K_1(1400)\to(K\pi_i)\pi_j} = EC_{(K\pi_i)\pi_j} g_{1400,D}^{K\pi_i} D^{(K\pi_i\pi_j)}(-\cos\theta_{K_1} + \sqrt{2}\sin\theta_{K_1}) \quad (70)$$

$$\tilde{A}_S^{K_1(1270)\to K(\pi_i\pi_j)} = EC_{K(\pi_i\pi_j)} g_{1270,S}^{\pi_i\pi_j} S^{(K\pi_i\pi_j)}(\sqrt{2}\sin\theta_{K_1} + \cos\theta_{K_1}) \quad (71)$$

$$\tilde{A}_D^{K_1(1270)\to K(\pi_i\pi_j)} = EC_{K(\pi_i\pi_j)} g_{1270,D}^{\pi_i\pi_j} D^{(K\pi_i\pi_j)}(-\sin\theta_{K_1} + \sqrt{2}\cos\theta_{K_1}) \quad (72)$$

$$\tilde{A}_S^{K_1(1400)\to K(\pi_i\pi_j)} = EC_{K(\pi_i\pi_j)} g_{1400,S}^{\pi_i\pi_j} S^{(K\pi_i\pi_j)}(\sqrt{2}\cos\theta_{K_1} - \sin\theta_{K_1}) \quad (73)$$

$$\tilde{A}_D^{K_1(1400)\to K(\pi_i\pi_j)} = EC_{K(\pi_i\pi_j)} g_{1400,D}^{\pi_i\pi_j} D^{(K\pi_i\pi_j)}(-\cos\theta_{K_1} - \sqrt{2}\sin\theta_{K_1}) \quad (74)$$

In the original paper author indroduces only θ_{K_1} — mixing angle between non-mass eigenstates, but in order to enable flexibility of the fit 8 additional complex parameters are introduced: $g_{1270,S}^{K^*/\rho}, g_{1270,D}^{K^*/\rho}, g_{1400,S}^{K^*/\rho}, g_{1400,D}^{K^*/\rho}$.

1.3.2. *Hadronic form-factors*

Once amplitudes are calculated one can compute hadronic form-factors for 1^+:

$$A_{K^*}^{K_1(1270/1400)\to(K\pi_i)\pi_j} = \tilde{A}_S^{K_1(1270/1400)\to(K\pi_i)\pi_j} + \sqrt{2}\tilde{A}_D^{K_1(1270/1400)\to(K\pi_i)\pi_j}\exp(i\varphi_{K^*}^D) \quad (75)$$

$$A_\rho^{K_1(1270/1400)\to K(\pi_i\pi_j)} = \left(\tilde{A}_S^{K_1(1270/1400)\to K(\pi_i\pi_j)} + \sqrt{2}\tilde{A}_D^{K_1(1270/1400)\to K(\pi_i\pi_j)}\exp(i\varphi_\rho^D)\right)\exp(i\varphi_\rho^S) \quad (76)$$

$$\begin{aligned} B_{K^*}^{K_1(1270/1400)\to(K\pi_i)\pi_j} &= \left[\left[1 - \sqrt{\frac{s_{K\pi_i}}{E(s, s_{K\pi_i}, m_{\pi_j})}}\right]\tilde{A}_S^{K_1(1270/1400)\to(K\pi_i)\pi_j}\right. \\ &+ \left.\sqrt{2}\left[1 + \frac{2\sqrt{s_{K\pi_i}}}{E(s, s_{K\pi_i}, m_{\pi_j})}\right]\tilde{A}_D^{K_1(1270/1400)\to(K\pi_i)\pi_j}\exp(i\varphi_{K^*}^D)\right]\frac{E(s, s_{K\pi_i}, m_{\pi_j})}{\sqrt{s}p(s, s_{K\pi_i}, m_{\pi_j})^2} \end{aligned} \quad (77)$$

$$\begin{aligned} B_\rho^{K_1(1270/1400)\to K(\pi_i\pi_j)} &= \left[\left[1 - \sqrt{\frac{s_{\pi_i\pi_j}}{E(s, s_{\pi_i\pi_j}, m_K)}}\right](\tilde{A}_S^{K_1(1270/1400)\to K(\pi_i\pi_j)}\right. \\ &+ \left.\sqrt{2}\left[1 + \frac{2\sqrt{s_{\pi_i\pi_j}}}{E(s, s_{\pi_i\pi_j}, m_K)}\right]\tilde{A}_D^{K_1(1270/1400)\to K(\pi_i\pi_j)}\exp(i\varphi_\rho^D))\exp(i\varphi_\rho^S)\right] \\ &\times \frac{E(s, s_{\pi_i\pi_j}, m_K)}{\sqrt{s}p(s, s_{\pi_i\pi_j}, m_K)^2} \end{aligned} \quad (78)$$

Within the expressions three model parameters are inctroduced ($\varphi_S^\rho, \varphi_D^\rho, \varphi_D^{K^*}$). These parameters are called hadronic phases and they responsible for defining direction of S and D waves in the complex space.

1.3.3. *Helicity currents*

After calculation of hadronic form-factors one can obtain expressions for helicity currents:

- $C_{12}^{K^*(1270/1400)}$, $C_{12}^{\rho(1270/1400)}$:

$$\begin{aligned} C_1^{K^*(1270/1400)} &= CG_{K^*}g_{K^*K\pi}\Bigg[\Big[A_{K^*}^{K_1(1270/1400)\to(K\pi_2)\pi_1}(1+\frac{m_{\pi_1}^2-m_K^2}{s_{K\pi_2}})+B_{K^*}^{K_1(1270/1400)\to(K\pi_2)\pi_1} \\ &\times \Big(2p_{\pi_1}p_{\pi_2}-(1+\frac{m_{\pi_1}^2-m_K^2}{s_{K\pi_2}})(\sqrt{s}E_{\pi_1}-m_{\pi_1}^2)\Big)\Big]BW_{K^*}(s_{K\pi_2}) \\ &- 2A_{K^*}^{K_1(1270/1400)\to(K\pi_1)\pi_2}BW_{K^*}(s_{K\pi_1})\delta\Bigg] \end{aligned} \quad (79)$$

$$\begin{aligned} C_2^{K^*(1270/1400)} &= CG_{K^*}g_{K^*K\pi}\Bigg[\Big[A_{K^*}^{K_1(1270/1400)\to(K\pi_1)\pi_2}(1+\frac{m_{\pi_2}^2-m_K^2}{s_{K\pi_1}})+B_{K^*}^{K_1(1270/1400)\to(K\pi_1)\pi_2} \\ &\times \Big(2p_{\pi_1}p_{\pi_2}-(1+\frac{m_{\pi_2}^2-m_K^2}{s_{K\pi_1}})(\sqrt{s}E_{\pi_2}-m_{\pi_2}^2)\Big)\Big]BW_{K^*}(s_{K\pi_1}) \\ &- 2A_{K^*}^{K_1(1270/1400)\to(K\pi_2)\pi_1}BW_{K^*}(s_{K\pi_2})\delta\Bigg] \end{aligned} \quad (80)$$

$$C_1^{\rho(1270/1400)} = CG_\rho g_{\rho\pi\pi}\Big[A_\rho^{K_1(1270/1400)\to K(\pi_1\pi_2)}-B_\rho^{K_1(1270/1400)\to K(\pi_1\pi_2)}\sqrt{s}(E_{\pi_1}-E_{\pi_2})\Big]BW_\rho(s_{\pi_1\pi_2}) \quad (81)$$

$$C_2^{\rho(1270/1400)} = CG_\rho g_{\rho\pi\pi}\Big[A_\rho^{K_1(1270/1400)\to K(\pi_1\pi_2)}+B_\rho^{K_1(1270/1400)\to K(\pi_1\pi_2)}\sqrt{s}(E_{\pi_1}-E_{\pi_2})\Big]BW_\rho(s_{\pi_1\pi_2}) \quad (82)$$

where $CG_{K^{*892}}=\begin{cases}-\frac{2}{3}, & K_{res}^+\to K^+\pi^+\pi^-\\ \frac{\sqrt{2}}{3}, & K_{res}^0\to K^+\pi^0\pi^-\end{cases}$, $CG_\rho=\begin{cases}-\frac{1}{\sqrt{6}}, & K_{res}^+\to K^+\pi^+\pi^-\\ \frac{1}{\sqrt{3}}, & K_{res}^0\to K^+\pi^0\pi^-\end{cases}$ — Clebsh-Gordon coefficients,

$\delta=\begin{cases}0, K_{res}^+\to K^+\pi^+\pi^-\\ 1, K_{res}^0\to K^+\pi^0\pi^-\end{cases}$ — coefficient which acounts for decay of $K^{*892}\to K^+\pi^0$;

- $C_{12}^{(1270/1400)}$:

$$C_{1,2}^{(1270/1400)}=\Big[(C_{1,2}^{K^*(1270/1400)}+C_{1,2}^{\rho(1270/1400)})\Big]BW^{(1270/1400)}(s) \quad (83)$$

- Helicity currents of 1270 and 1400 can be written as:

$$\overrightarrow{J}{}^{\prime 1270/1400}=f^{1270/1400}\Big[C_1^{1270/1400}\overrightarrow{p}_{\pi_1}-C_2^{1270/1400}\overrightarrow{p}_{\pi_2}\Big] \quad (84)$$

where $f^{1270/1400}$ stands for coupling constants of 1270/1400 resonances.

1.4. Model of $1^-\to(1^-\to0^-0^-)0^-$ decays

- $C_3^{K^*(1410/1680)}$, $C_3^{\rho(1410/1680)}$:

$$C_3^{K^*(1410/1680)} = 2i\sqrt{s}CG_{K^*}g_{K^*K\pi}\tilde{g}_{K^*}^{1410/1680}\Big[BW_{K^*}(s_{K_{\pi_2}})+\delta\cdot BW_{K^*}(s_{K\pi_1})\Big] \quad (85)$$

$$C_3^{\rho(1410/1680)} = 2i\sqrt{s}CG_\rho g_{\rho\pi\pi}\tilde{g}_\rho^{1410/1680}BW_\rho(s_{\pi_1\pi_2}) \quad (86)$$

- $C_3^(1410/1680)$:

$$C_3^{1410/1680}=\Big[C_3^{K^*(1410/1680)}+C_3^{\rho(1410/1680)}\Big]BW^{1410/1680}(s) \quad (87)$$

- Helicity currents of 1410 and 1680 can be expressed as:

$$\overrightarrow{L}{}^{\prime 1410/1680}=f^{1410/1680}C_3^{1410/1680}[\overrightarrow{p}_{\pi_1}\times\overrightarrow{p}_{\pi_2}] \quad (88)$$

1.5. Model of $2^+\to(1^-\to0^-0^-)0^-$ decays

- $C_{4,5}^{K^*(1430)}$, $C_4^{\rho(1430)}$:

$$C_4^{K^*(1430)} = \sqrt{2s}iCG_{K^*}g_{K^*K\pi}\tilde{g}_{K^*}^{1430}BW_{K^*}(s_{K\pi_2}) \quad (89)$$

$$C_5^{K^*(1430)} = \sqrt{2s}iCG_{K^*}g_{K^*K\pi}\tilde{g}_{K^*}^{1430}BW_{K^*}(s_{K\pi_1}) \quad (90)$$

$$C_4^{\rho(1430)} = \sqrt{2s}iCG_\rho g_{\rho\pi\pi}\tilde{g}_\rho^{1430}BW_{K^*}(s_{\pi_1\pi_2}) \quad (91)$$

- $C_{4,5}^{1430}$:

$$C_{4,5}^{1430} = f^{1430}(m_B^2 - s)\Big[C_4^{\rho(1430)} \mp C_4^{K^*(1430)}\Big]BW^{1430}(s) \tag{92}$$

$$\tag{93}$$

- Helicity current of 1430:

$$\begin{aligned}\overrightarrow{K}'^{1430} &= C_4^{1430}\Big[(\overrightarrow{\varepsilon}_0\cdot\overrightarrow{p}_{\pi_1})[\overrightarrow{p}_{\pi_1}\times\overrightarrow{p}_{\pi_2}] + \overrightarrow{p}_{\pi_1}(\overrightarrow{\varepsilon}_0\cdot[\overrightarrow{p}_{\pi_1}\times\overrightarrow{p}_{\pi_2}])\Big] \\ &+ C_5^{1430}\Big[(\overrightarrow{\varepsilon}_0\cdot\overrightarrow{p}_{\pi_2})[\overrightarrow{p}_{\pi_1}\times\overrightarrow{p}_{\pi_2}] + \overrightarrow{p}_{\pi_2}(\overrightarrow{\varepsilon}_0\cdot[\overrightarrow{p}_{\pi_1}\times\overrightarrow{p}_{\pi_2}])\Big]\end{aligned} \tag{94}$$

1.6. Model of $2^- \to (1^- \to 0^-0^-)0^-$ decays

$$C_{\pi_1\pi_2} = 2\Big(\big(f_V + h_V\sqrt{s} * (E_{\pi_1} + E_{\pi_2})\big)(m_{\pi_1}^2 - m_{\pi_2}^2)/s_{\pi_1\pi_2} - h_V\sqrt{s}(E_{\pi_1} - E_{\pi_2})\Big) \tag{95}$$

$$C_{K\pi_1} = 2\Big(\big(f_V + h_V\sqrt{s}(E_{\pi_1} + E_K)\big)(m_{\pi_1}^2 - m_K^2)/s_{K\pi_1} - h_V\sqrt{s}(E_{\pi_1} - E_K)\Big) \tag{96}$$

$$C_{K\pi_2} = 2\Big(\big(f_V + h_V\sqrt{s}(E_{\pi_2} + E_K)\big)(m_{\pi_2}^2 - m_K^2)/s_{K\pi_2} - h_V\sqrt{s}(E_{\pi_2} - E_K)\Big) \tag{97}$$

$$K_{\pi_1\pi_2} = i\sqrt{\frac{2}{s}}CG_\rho BW_{\rho^{775}}(s_{\pi_1\pi_2}) \tag{98}$$

$$K_{K\pi_1} = i\sqrt{\frac{2}{s}}CG_{K^*}BW_{K^{*892}}(s_{K\pi_1}) \tag{99}$$

$$K_{K\pi_2} = i\sqrt{\frac{2}{s}}CG_{K^*}BW_{K^{*892}}(s_{K\pi_2}) \tag{100}$$

$$\tag{101}$$

$$c_{\pi_1\pi_2} = K_{\pi_1\pi_2}\Big(C_{\pi_1\pi_2}\big((\overrightarrow{\varepsilon}_0\cdot\overrightarrow{p}_{\pi_1}) + (\overrightarrow{\varepsilon}_0\cdot\overrightarrow{p}_{\pi_2})\big) - 2f_V^\rho(\overrightarrow{\varepsilon}_0\cdot\overrightarrow{p}_{\pi_1})\Big) \tag{102}$$

$$c_{K\pi_1} = \delta K_{K\pi_1}\Big(f_V^{K^*}(\overrightarrow{\varepsilon}_0\cdot\overrightarrow{p}_{\pi_1}) + 2C_{K\pi_1}\big((\overrightarrow{\varepsilon}_0\cdot\overrightarrow{p}_{\pi_1}) + (\overrightarrow{\varepsilon}_0\cdot\overrightarrow{p}_{\pi_2})\big)\Big) \tag{103}$$

$$c_{K\pi_2} = K_{K\pi_2}f_V^{K^*}(\overrightarrow{\varepsilon}_0\cdot\overrightarrow{p}_{\pi_2}) \tag{104}$$

$$C_6 = c_{\pi_1\pi_2} + c_{K\pi_1} + c_{K\pi_2} \tag{105}$$

$$\overrightarrow{M} = f^{1580}(C_6\overrightarrow{p_{\pi_1}} - C_7\overrightarrow{p_{\pi_2}}) \tag{106}$$

1.7. Model of $2^- \to (1^- \to 0^-0^-)0^-$ decays

1.8. *Decay rate*

Radiative $b \to s\gamma$ transition ocuurs either through $b \to s_L\gamma_L$ or $b \to s_R\gamma_R$. Since the final states are different, differential decay rate is the sum of two decay rates:

$$\frac{d\Gamma}{d\cos\theta d\varphi ds_{K\pi_1}ds_{\pi_1\pi_2}ds} = \Big[\frac{d\Gamma}{d\cos\theta d\varphi ds_{K\pi_1}ds_{\pi_1\pi_2}ds}\Big]_L + \Big[\frac{d\Gamma}{d\cos\theta d\varphi ds_{K\pi_1}ds_{\pi_1\pi_2}ds}\Big]_R \tag{107}$$

Therefore additional modifications to the original helicity currents of $K_{res} \to K\pi\pi$ decay should be applied:

$$\overrightarrow{J}^{1270}_{L,R} = \overrightarrow{J}'^{1270} \tag{108}$$

$$\overrightarrow{J}^{1400}_{L,R} = \frac{m_B^2 - m^2_{K_1^{1400}}}{m_B^2 - m^2_{K_1^{1270}}} \overrightarrow{J}'^{1400} \tag{109}$$

$$\overrightarrow{L}^{1410}_{L,R} = \pm \left[\frac{m_B^2 - m^2_{K_1^{1410}}}{m_B^2 - m^2_{K_1^{1270}}} \right]^{\frac{3}{2}} \overrightarrow{L}'^{1410} \tag{110}$$

$$\overrightarrow{L}^{1680}_{L,R} = \pm \left[\frac{m_B^2 - m^2_{K_1^{1680}}}{m_B^2 - m^2_{K_1^{1270}}} \right]^{\frac{3}{2}} \overrightarrow{L}'^{1680} \tag{111}$$

$$\overrightarrow{K}^{1430}_{L,R} = \pm \left[\frac{m_B^2 - m^2_{K_1^{1430}}}{m_B^2 - m^2_{K_1^{1270}}} \right]^{\frac{3}{2}} \overrightarrow{K}'^{1430} \tag{112}$$

$$\tag{113}$$

Then differential decay rate can be expressed as following:

$$\begin{aligned} \frac{d\Gamma}{d\cos\theta d\varphi ds_{K\pi_1} ds_{\pi_1\pi_2} ds} &= \frac{1-\lambda}{2} \left| \overrightarrow{\varepsilon}_L \cdot (\overrightarrow{J}^{1270}_L + \overrightarrow{J}^{1400}_L + \overrightarrow{L}^{1410}_L + \overrightarrow{L}^{1680}_L + \overrightarrow{K}^{1430}_L) \right|^2 \\ &+ \frac{1+\lambda}{2} \left| \overrightarrow{\varepsilon}_R \cdot (\overrightarrow{J}^{1270}_R + \overrightarrow{J}^{1400}_R + \overrightarrow{L}^{1410}_R + \overrightarrow{L}^{1680}_R + \overrightarrow{K}^{1430}_R) \right|^2 \end{aligned} \tag{114}$$

where λ — photon polarization parameter.

2. Generator

Decay rate produced by above theoretical model is a fucntion of phase space and model parameters. The latters remain unchanged during events generation process, while phase space is sampled from allowed kinematical region. The events generation process of 5D data is the following:

Algorithm 2: Events sampling from the p.d.f

Result: $\mathbf{\Omega}$
$\mathbf{\Omega} = []$
Find maximum decay rate (DR_{max}) value using hill climbing algorithm;
while *($i_{event} < N_{events}$)* **do**
  $\overrightarrow{\Omega}_{i_{event}} = (m^2_{K\pi\pi}, m^2_{\pi\pi}, m^2_{K\pi}, \cos\theta, \varphi)$ =*Generate Phase Space()*;
  $DR_{i_{event}} = CalculateDecayRate(\overrightarrow{\Omega}_{i_{event}} | \overrightarrow{\alpha})$;
  $u_i = uniform(0, DR_{max})$;
  if $DR_{i_{event}} > u_i$ **then**
    Append $\overrightarrow{\Omega}_{i_{event}}$ to $\mathbf{\Omega}$;
    Increase counter i_{event} by unit;
  end
end

where $\mathbf{\Omega}$ — phase space matrix with rows represented phase space vectors $\overrightarrow{\Omega}_{i_{events}}$, which passed hit-or-miss condition. Interface for using the generator in user code is the following:

- Create generator object with specifying the type of decay (either $B \to K^+\pi^+\pi^-\gamma$ or $B \to K^+\pi^0\pi^-\gamma$) by changing parameter *charge*;
- Set path to the file with model parameters;
- Generate events specifying number of events;
- Write results to root file;

File with model parameters looks like dictionary with key-value pairs. Where key — name of model parameter, value — value of corresponding parameter. Output of the generator is a root file containing 4-momenta of each final state particle for each generated event. Generator relies on:

- *GInterfaceForMinimization.cxx/GInterfaceForMinimization.h* — sets function, variabes and parameters which will be used as input for generator core. It is also responsible for looking up the local maximum of decay rate using hill climbing algorithm;
- *GEventsGenerator.cxx/GEventsGenerator.h* — the core of events generation. Here event's decay rates are calculated according to the Algorithms 1-2.
- *GGenerator.cxx/GGenerator.h* — interface between user and generator engine, which defines functionality for setting up the generator, model couplings, events number and calls the Algorithm 2;
- *GGeneratorParser.cxx/GGeneratorParser.h* — utility parser of **.ge* files, responsible for extracting (key: value) pairs from the file.

Example to reproduce:

Let us generate phase space distributed according to the probability density function of $K_1^{1270}, K_1^{1400}, K^{*1410}, K^{*1680}, K_2^{1430}$. The following configuration file has to be specified (the rest of parameters assigned to its default values):

```
thetaK1:            1.042; # mixing angle of K1(1270) and K1(1400)
phiDK*:             0.; # hadronic phase of K*(892) in K1
phiSRho:            0.; # hadronic phase of S-wave of rho(770) in K1
phiDRho:            0.; # hadronic phase of D-wave of rho(770) in K1
lambda:             0.7; # photon polarization parameter
ffRe_1400:          0.47; # K1(1400) coupling (real part)
ffIm_1400:          0.; # K1(1400) coupling (imaginary part)
gg3Kstr_1410:       1.; # K*(892) coupling in K*(1410), real part
gg3ImKstr_1410:     0.; # K*(892) coupling in K*(1410), imaginary part
gg3Rho_1410:        1.; # rho(770) coupling in K*(1410), real part
gg3ImRho_1410:      0.; # rho(770) coupling in K*(1410), imaginary part
gg4Kstr_1680:       1.; # K*(892) coupling in K*(1680), real part
gg4ImKstr_1680:     0.; # K*(892) coupling in K*(1680), imaginary part
gg4Rho_1680:        1.; # rho(770) coupling in K*(1680), real part
gg4ImRho_1680:      0.; # rho(770) coupling in K*(1680), imaginary part
gg5Kstr_1430:       1.; # K*(892) coupling in K2(1430), real part
gg5ImKstr_1430:     0.; # K*(892) coupling in K2(1430), imaginary part
gg5Rho_1430  :      1.; # rho(770) coupling in K2(1430), real part
gg5ImRho_1430:      0.; # rho(770) coupling in K2(1430), imaginary part
```

Listing 1. Configuration file of generator (model.ge)

Below sample script is given:

```
#include "GGenerator.h"

using namespace Gamapola;

int main()
{
    auto g1 = std::make_shared<GGenerator>(0); // generator object for neutral B-meson decay
    g1->GSetCouplings("../additionalFiles/model.ge"); // set up the couplings from the configuration file
    g1->GGenerate(1000); // set up number of events
    g1->GWriteToFile("neutral.root"); // perform events generation and write results to the root-file

    return 0;
}
```

Listing 2. User code of generator

After compiling and running the output root-file is produced. It has the following structure:

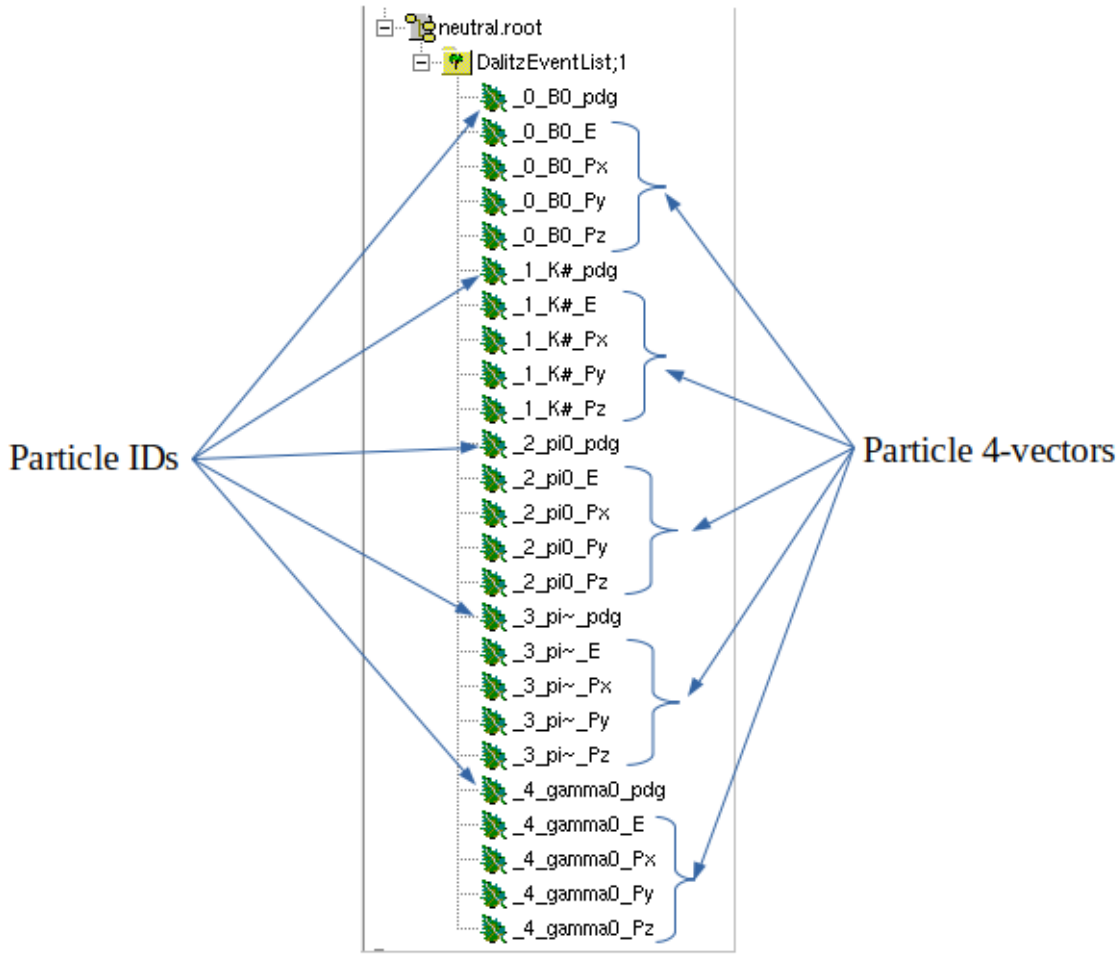
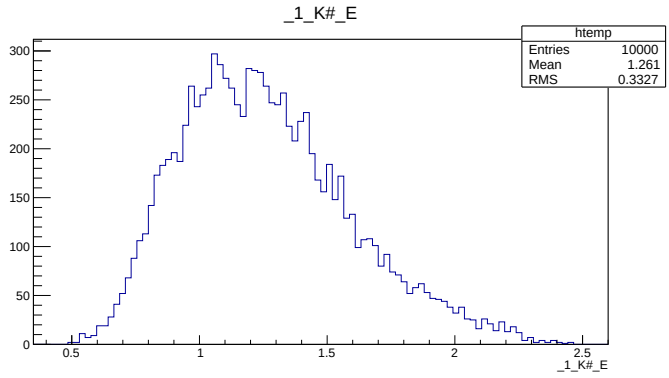
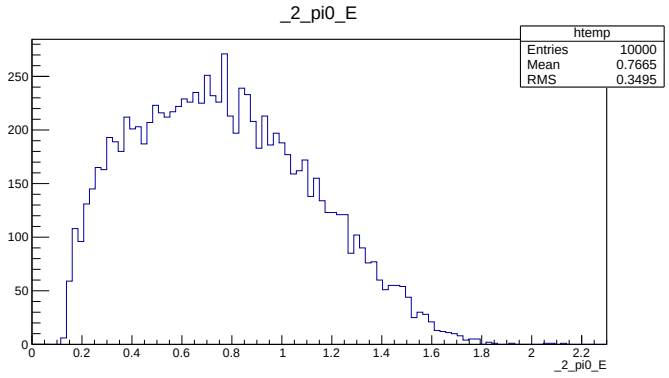


Figure 1. Output root-file structure

Produced 4-vectors are in the B-rest frame.





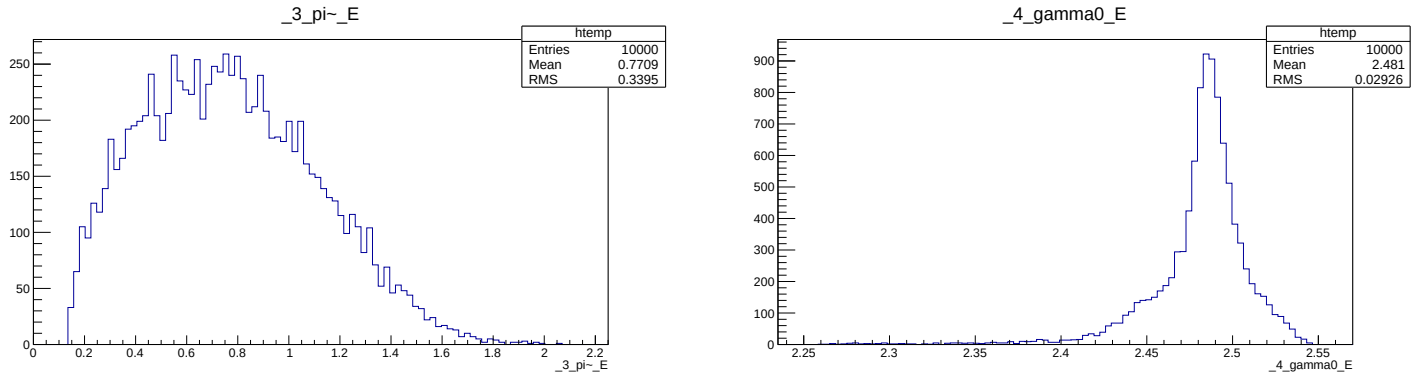


Figure 2. Energies of final state particles in the B-rest frame

Histograms drawing:
In order to obtain phase space distributions the following script can be used:

```
#include "GamPolaPlotter/GGampolaPlotter.h"

using namespace Gamapola;

int main()
{
    auto&& plotter = std::make_shared<GGampolaPlotter>("../additionalFiles/plot.img"); // create plotter object with specifying drawing options in plot-file
    plotter->plot("neutral.root", 0, "neutral_plot.root"); // specify input root file, charge and output root-file with histograms
    return 0;
}
```

Listing 3. User code of generator

Configuration file for plotting projections of phase space contains:

```
#dimensionality, range flag: label:            type,   bins, low, up
1D,              All:        sKpipi:           sKpipi,  100,  1,   2
1D,              All:        sKpi1:            sKpi1,   100,  0,   2
1D,              All:        cosTheta:         cosTheta, 100,  -1,  1
1D,              All:        cosThetaBins10:   cosTheta, 10,  -1,  1

#dimensionality, range flag:             label:         type,     bins, low, up
1D,              cut: 1 < sKpipi < 1.4:  cosTHetaCut:   cosTheta, 100,  -1,  1

#dimensionality,    range flag:    label:          type,         binsX,   lowX,     upX,      binsY,    lowY,       upY
2D,                 All:           sKpi1Kpi2:      sKpi1Kpi2,    100,     0.4,      1.4,      100,      0.4,        1.4
2D,                 All:           sKpi1Kpi210x10: sKpi1Kpi2,    10,      0.4,      1.4,      10,       0.4,        1.4
```

Listing 4. Configuration flag for plotting (plot.img)

In general, three options are possible:

- One dimensional plotting in whole rangle of $M_{K\pi\pi}$;
- One dimensional plotting in specified range of $M_{K\pi\pi}$;
- Two dimensional plotting in whole rangle of $M_{K\pi\pi}$;

In the confiuration file above:

- 1st column is dimensionality of plotted distributions;
- 2nd — cut-flag, which stands for cuts in $M_{K\pi\pi}$ while building the histograms (only cut in one dimension is available);
- 3rd — label, the name given to the distribution. It can be any name;
- 4th — type of distribution has predefined values: *cosTheta, sKpipi, sKpi1, etc*
- 5th — number of bins for building the histogram;
- 6th — low limit when building the histogram;

- 7th — up limit when building the histogram;

As result the following distributions of phase space can be obtained as specified in *plot.img*:

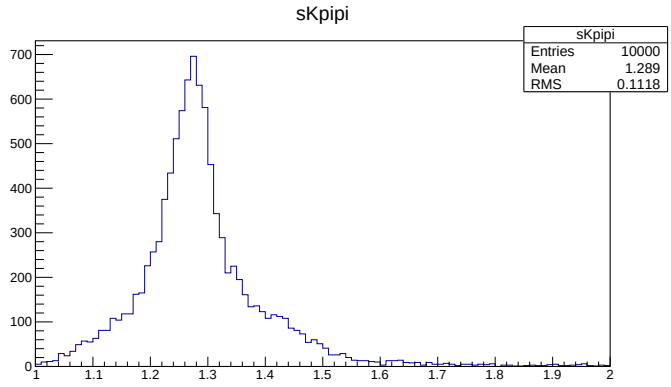


Figure 3. $M_{K\pi\pi}$

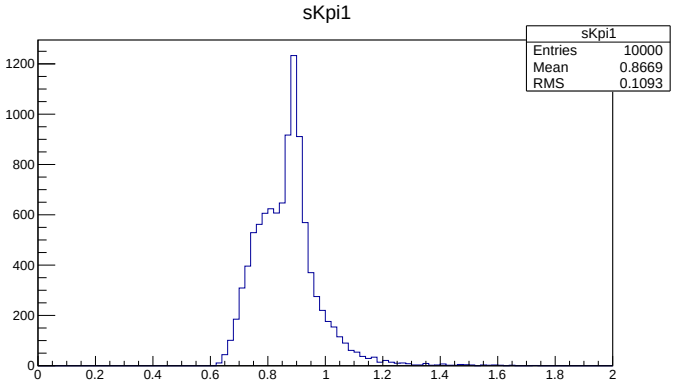


Figure 4. $s_{K\pi_1}$

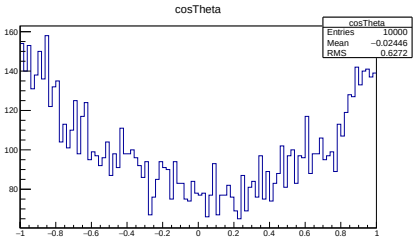


Figure 5. $\cos\theta$

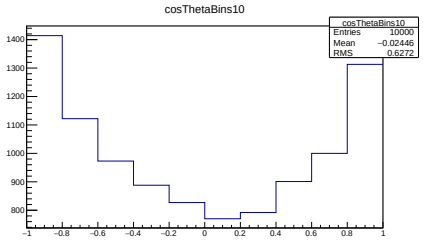


Figure 6. $\cos\theta$ with 10 bins

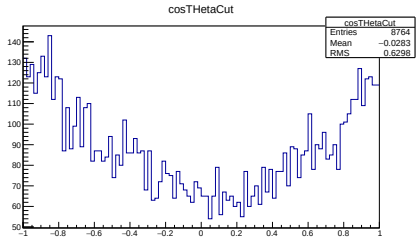


Figure 7. $1 < M_{K\pi\pi} < 1.4\ GeV$

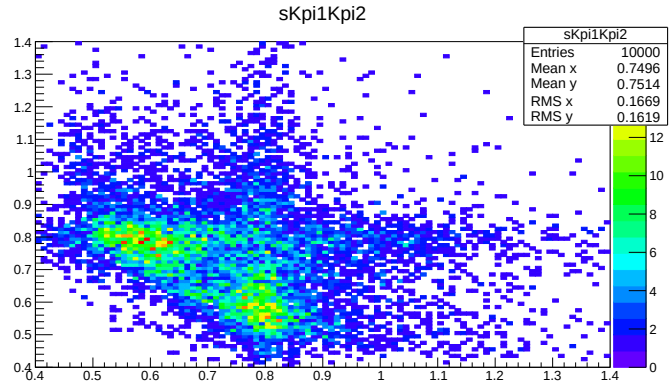


Figure 8. $s_{K\pi 1}$ vs $s_{K\pi_2}$

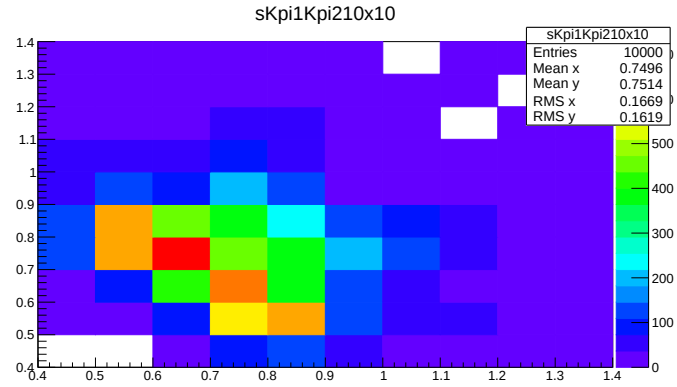


Figure 9. $s_{K\pi 1}$ vs $s_{K\pi_2}$ with 10 bins

2.1. Notes about EvtGen

In order to integrate the model in the EvtGen framework, one has to follow special conventional pattern of user-models insertion. It is achieved by inheriting parent's class *EvtGenAmplitude*. It allows to keep all public and protected methods of parent class, so new model class behaves as parent's one. In particular *EvtGamPolaBToKpipiGamma* class has to implement the following methods apart from constructor, destructor, clone[4]:

- init() — memory allocation, setting model parameters, mode;
- getName() — set name of the model;
- initProbMax() — set maximum value of decay rate for the given set of parameters;

[4]which means that developer of new model has to provide the following information

- decay(EvtParticle *p) — is executed for each loop iteration, where p stands for B-meson in our particular case;

The way “EvtGen” generates events is similar to the one used in “GamPola”:

- Set maximum decay rate value in the initialization step;
- Use hit-or-miss technique (EvtGen performs 10000 tries to get event, while GamPola has no limit on number of tries);

For clarity the pseudo code for decay looks as follwoing:

```
decay
{
  1) Create daughters of B meson decay according to the .DEC files
  2) Get these daughters
  3) Using GamPola, generate phase space with kinematical restrictions only
  4) Substitute phase space point to the decay rate expression
  5) Calculate PDF using decay rate
  6) amplitude = sqrt(PDF)
  7) send amplitude to vertex
  8) send 4-vectors to daughters
}
```

Listing 5. Decay function from EvtGen

For completeness *.DEC* file is given:

```
#tion of dm still in some decay models
Define dm 0.507e12
Define lambdaPhoton 0.95
Alias gammaR       gamma


Decay Upsilon(4S)
0.5000 B-sig B+              VSS;
0.5000 B+sig B-              VSS;
Enddecay


Decay B+sig

1.0 gammaR       K+      pi-     pi+      GAMPOLAR
######################## photon polarization:
     lambdaPhoton
######################## gammaQPCM & f2:
     4.           4.
######################## strong phases (thetaK1 & phiDK* & phiSRho & phiDRho & kappa):
     1.0472       0.     0.    0.      0.0        0.0
################################### K1_1400 (coupling)#######################:
     0.          0.0
################################### K'*0 (coupling)##########################:
     0.         0.0
     ################################### K*0 (892)(couplings)##############:
          1.0          0.0
     ################################### rho(775) (couplings)##############:
          1.0          0.0
################################### K''*0(coupling)##########################:
     0.          0.0
     ################################### K*0 (892)(couplings)##############:
          1.0          0.0
     ################################### rho(775)(couplings)##############:
          1.0          0.0
################################### K_2*0 (coupling)#########################:
     0.         0.0
     ################################### K*0 (892)(coupling)###############:
          1.0          0.0
     ################################### rho(775)(coupling)################:
          1.0          0.0;
Enddecay
#CDecay B-sig

Decay B-sig
Enddecay

End
```

Listing 6. Configuration decay file from EvtGen

The core of this file is the follwoing string:

```
1.0 gammaR        K+       pi-      pi+       GAMPOLAR
```

Listing 7. Core string

where:

- first entity — decay ratio;
- from second to fifth entity — final state particles of this particular decay;
- sixth entity — model name;

3. Fitter

Fitter utilies functions from the theoretical framework for calculating decay rates. It also calculates logarithmic likelihood for each optimiation step as a part of gradient descent algorithm. Fitting requires decay rate factorization:

$$\frac{d\Gamma}{d\cos\theta d\varphi ds_{K\pi\pi} ds_{K\pi_1} ds_{\pi_1\pi_2}} = \frac{d\Gamma}{d\overrightarrow{\Omega}} = F(\overrightarrow{\Omega}, \overrightarrow{\alpha}) = \sum_{i=0}^{N_{terms}} f_i(\overrightarrow{\Omega}) \cdot g_i(\alpha) \tag{115}$$

where N_{terms} — number of terms within decay rate expression, $f_{i,i}$ — i-th term within decay rate, which depends only on kinematical variables, g_i — i-th term within decay rate, which depends only on model parameters, $\overrightarrow{\Omega}$ — point in 5D phase space, $\overrightarrow{\alpha}$ — set of model parameters, F — differential decay rate;
After dividing decay rate function on its integral over allowed phace space defined by Algorithm 1 and fixed set of model parameters the p.d.f expression is obtained:

$$p.d.f = \frac{F(\overrightarrow{\Omega}, \overrightarrow{\alpha})}{\int_{\Omega_0} \sum_{i=0}^{N_{terms}} f_i(\overrightarrow{\Omega}) \cdot g_i(\overrightarrow{\alpha}) d\overrightarrow{\Omega}} = \frac{F(\overrightarrow{\Omega}, \overrightarrow{\alpha})}{\sum_{i=0}^{N_{terms}} \left[\int_{\overrightarrow{\Omega}_0} f_i(\overrightarrow{\Omega}) d\Omega \right] \cdot g_i(\overrightarrow{\alpha})} \tag{116}$$

On practice it means summation of the decay rates. It is also assumed that points of phase space are uniformly distributed.

$$\int_{\Omega_0} f_i(\overrightarrow{\Omega}) d\overrightarrow{\Omega} = \lim_{N\to\infty} \sum_{j=0}^{N} f_i(\overrightarrow{\Omega}_j) \frac{\Omega_0}{N} \approx \sum_{j=0}^{N_\Sigma} f_i(\overrightarrow{\Omega}_j) \frac{\Omega_0}{N_\Sigma} \tag{117}$$

where $\overrightarrow{\Omega}_0 = \int d\overrightarrow{\Omega}$ — integrated region of allowed phase space, N_Σ — total number of events used for calculation of noramlization integrals. In the above formulas i — refers to term number withing decay rate, j — refers to event number used for calculation normalization sum.
Then p.d.f is as following:

$$p.d.f = \frac{N_\Sigma}{\Omega_0} \frac{F(\overrightarrow{\Omega}, \overrightarrow{\alpha})}{\sum_{i=0}^{N_{terms}} \left[\sum_{j=0}^{N_\Sigma} f_i(\overrightarrow{\Omega}_j) \right] \cdot g_i(\overrightarrow{\alpha})} = \frac{N_\Sigma}{\Omega_0} \frac{F(\overrightarrow{\Omega}, \overrightarrow{\alpha})}{\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\overrightarrow{\alpha})} \tag{118}$$

where $\tilde{f}_i = \left[\sum_{j=0}^{N_\Sigma} f_i(\overrightarrow{\Omega}_j) \right]$ — normalization integrals.
Likelihood function in general form can be written as:

$$L = \prod_{e=0}^{N_e} p.d.f(\overrightarrow{\Omega}_e | \overrightarrow{\alpha}) = \left[\frac{N_\Sigma}{\Omega_0} \right]^{N_e} \prod_{e=0}^{N_e} \frac{F(\overrightarrow{\Omega}_e, \overrightarrow{\alpha})}{\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\overrightarrow{\alpha})} \tag{119}$$

where N_e — number of events used for estimating $\overrightarrow{\alpha}$, $\overrightarrow{\Omega}_e$ — point in the 5D phase space. After logarithmization:

$$-2\log\mathcal{L} = -2N_e \log\frac{N_\Sigma}{\Omega_0} - 2\sum_{e=0}^{N_e} \log F(\overrightarrow{\Omega}_e, \overrightarrow{\alpha}) + 2N_e \log\left[\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\overrightarrow{\alpha}) \right] \tag{120}$$

The first term of Eq. 120 is constant with respect to $\overrightarrow{\alpha}$ and can be removed:

$$obj.func = -2\sum_{e=0}^{N_e} \log F(\overrightarrow{\Omega}_e, \overrightarrow{\alpha}) + 2N_e \log\left[\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\overrightarrow{\alpha}) \right] \tag{121}$$

This function contains two terms:

- $-2\sum_{e=0}^{N_e} \log F(\overrightarrow{\Omega}_e, \alpha)$ — computation of decay rate (see Eq. 114);
- $2N_e \log\left[\sum_{i=0}^{N_{terms}} \tilde{f}_i \cdot g_i(\alpha) \right]$ — normalization part, which is computed by other means. In particular N_{terms}, $\tilde{f}_i$, $g_i(\alpha)$ should be known.

3.1. Fitting

Fitter works in the two regimes:

- Caching — stage, where normalization integrals $\tilde{f}_i$ are calculated and correspondings of these integrals and couplings $g_i(\overrightarrow{\alpha})$ are created (done by creating dynamic library). In this stage fitter produces:
  - *.gar* files, which specify symbolic correspondance between kinematical terms and model terms within decay rate (see Eqn. 115);
  - *.cxx* files — represent the source code of implemented $\tilde{f}_i$ and α_i;
  - *.so* files — dynamic libraries, which intended to perform fast calls of precompiled function, which recalculates $g_i(\overrightarrow{\alpha})$ terms, for each step of minimization algorithm;
  - *.txt* — file with normalization integrals $\tilde{f}_i$;

  For example, user code for caching step looks as following:

```
#include "GFitter.h"

using namespace Gamapola;

int main()
{
    auto f = std::make_shared<GFitter>(0); // create fitter object
    f->GGenerateNormalizationIntegrals("norm_neutral.txt"); // calculate normalization integrals

  return 0;
}
```

Listing 8. User code of caching step

- Fitting — using precompiled libraries, normalization integrals obtained on caching step, perform objective function minimization (gradient descent fully relies on ROOT minimizer).

Fitter relies on following classes:

- *GInterfaceForMinimization* — sets function, variabes and parameters which will be used as input for fitter core.
- *GEventsAnalyzer* — the core of fitting part, uses minuit minimizer to perform gradient descent, calculates normalization integrals;
- *GFitter* — interface between user and fitter engine, which defines functionality for setting up the function for optimization, model couplings and variables;
- *GSymbolicExpressionsAnalyzer* — responsible for defining the process of $\tilde{f}_i$ calculation;
- *GExCompiler* — responsible for creation, compilation and linking of *.cxx, .so, .gar* files;
- *GSymbolicMathFunctions* — responsible for defyning the structure of symbolic decay rate;
- *GFitterParser* — utility class used for parsing *.fit* files and extraction the specification of model parameters to estimate;
- *.fit* files — provide parameter's description and specification. Represent key, tuple pairs, where key — name of parameter (the same as in *.ge* file), tuple — collection of objects of different types such as *initial value, type, low limit, up limit and minimization step*;

Example of user code for fitting step is given below:

```
#include "GFitter.h"

using namespace Gamapola;

int main()
{
    auto f = std::make_shared<GFitter>(0); // create fitter object with neutral decay mode
    f->GSetModelParameters("../additionalFiles/model.fit");  // specify configuration file with model parameters
    f->GSetNormalizationIntegrals("norm_neutral_5k.txt"); // specify normalization integrals obtained on caching step
    f->GSetData("neutral.root", "1 < sKpipi < 1.8"); // set data and cut where to fit
    f->GFit(true); // fit the data and perform goodness of fit
    return 0;
}
```

Listing 9. User code of fitting step

Goodness of fit is performed using Kolmogorov-Smirnov unbined test. Using above mentioned script as a core one can perform sensitivity studies. The model used for generation of events for all cases contains: $K_1^{1270}, K_1^{1400}, K^{*1410}, K^{*1680}, K_2^{1430}$.

Fitting stage produces table in the *.csv* file which has the following structure:

Min val	fit status	GOF results	*pars*	*parerrs*

Table 2. Output format of the fitter

3.2. Sensitivity studies by resonance

By switching on/off resonances (varying their couplings) one can try to fit above mentioned data. Consider two cases:

- Full model with $K_1^{1270}, K_1^{1400}, K^{*1410}, K^{*1680}, K_2^{1430}$ resonances:

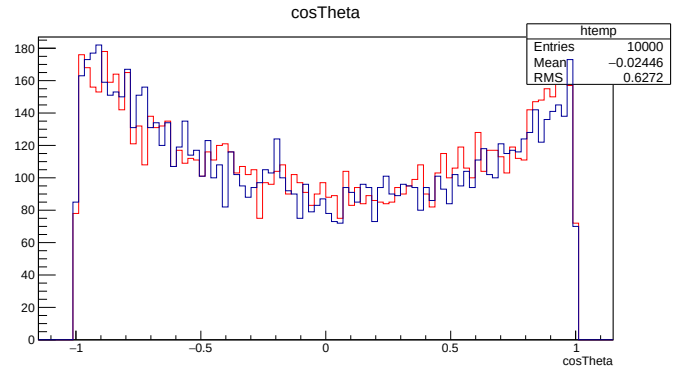


Figure 10. $\cos\theta$: $pValue = 0.0710474$

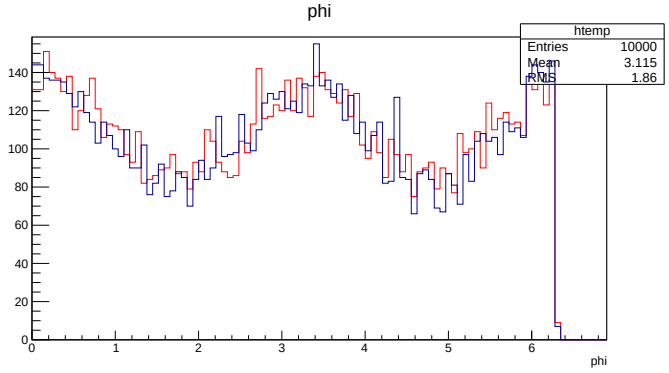


Figure 11. φ: $pValue = 0.556866$

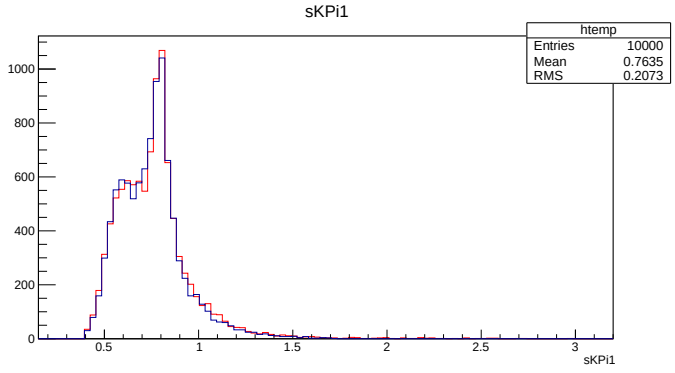


Figure 12. $s_{K\pi_1}$: $pValue = 0.157153$

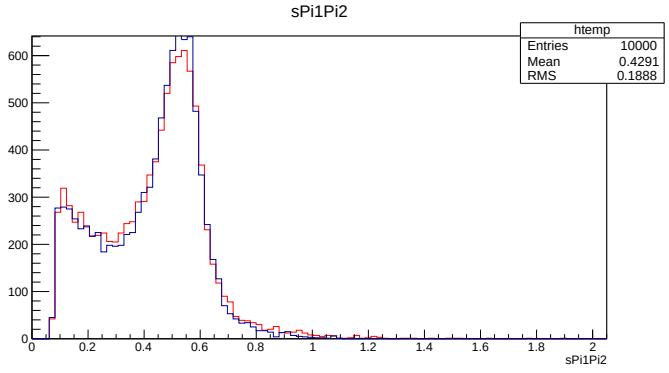


Figure 13. $s_{\pi_1\pi_2}$: $pValue = 0.0415591$

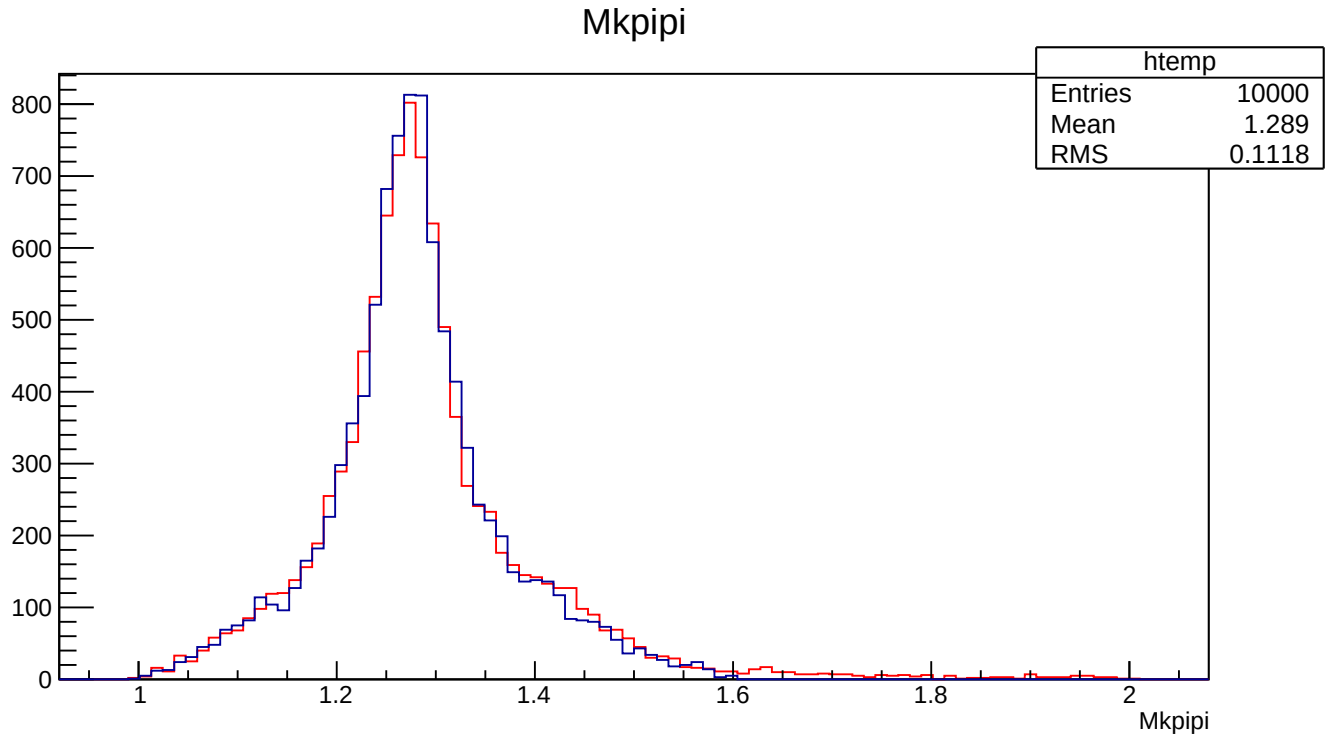


Figure 14. $M_{K\pi\pi}$: $pValue = 0.129104$

```
thetaK1         :    truth:    1.042     fit:      1.033e+00    error:     9.837e-03
phiDK*          :    truth:    0.000     fit:      6.332e+00    error:     5.965e-02
phiSRho         :    truth:    0.000     fit:      6.357e-02    error:     3.307e-02
phiDRho         :    truth:    0.000     fit:     -5.864e-01     error:     2.472e-01
lambda          :    truth:    0.700     fit:      7.283e-01    error:     3.475e-02
ffRe_1400       :    truth:    0.470     fit:      4.457e-01    error:     1.679e-02
ffIm_1400       :    truth:    0.000     fit:     -3.285e-03     error:     2.288e-02
gg3Kstr_1410    :    truth:    1.000    fit:     1.096e+00    error:     2.504e-01
gg3ImKstr_1410  :    truth:    0.000    fit:     -7.899e-01    error:     3.215e-01
gg3Rho_1410     :    truth:    1.000     fit:      1.657e+00    error:     5.340e-01
gg3ImRho_1410   :    truth:    0.000     fit:     -8.334e-01     error:     5.952e-01
gg4Kstr_1680    :    truth:    1.000     fit:     -2.764e-01     error:     1.924e-01
gg4ImKstr_1680  :    truth:    0.000     fit:     -1.632e-01     error:     2.170e-01
gg4Rho_1680     :    truth:    1.000     fit:     -3.649e-01     error:     3.525e-01
gg4ImRho_1680   :    truth:    0.000     fit:     -3.217e-01     error:     3.095e-01
gg5Kstr_1430    :    truth:    1.000     fit:      8.479e-01    error:     3.708e-02
gg5ImKstr_1430  :    truth:    0.000     fit:      1.371e-02    error:     5.440e-02
gg5Rho_1430     :    truth:    1.000     fit:      7.497e-01    error:     6.768e-02
gg5ImRho_1430   :    truth:    0.000     fit:      1.010e-01    error:     8.270e-02
```

Listing 10. Table of fit parameters using full model

- Reduced model with only K_1^{1270}:

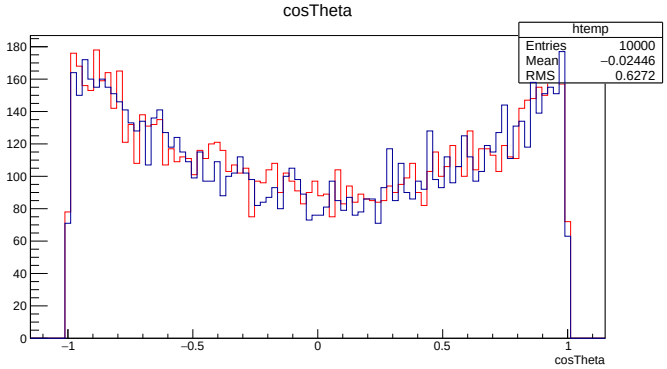


Figure 15. $\cos\theta$: $pValue = 0.883206$

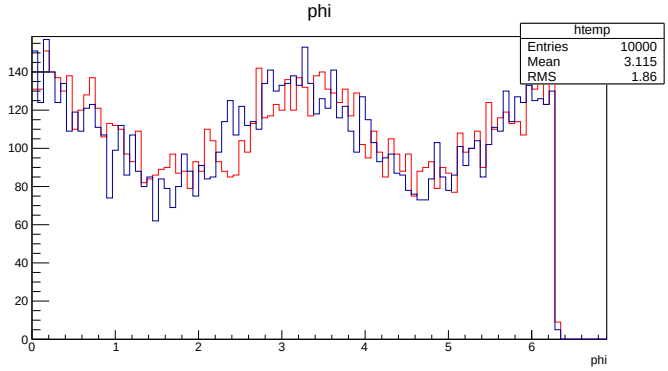


Figure 16. φ: $pValue = 0.133473$

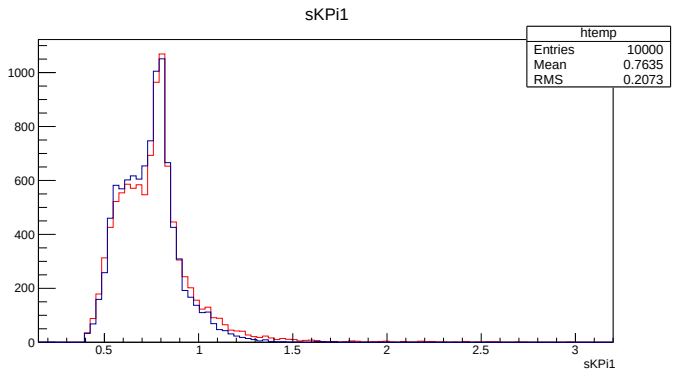


Figure 17. $s_{K\pi_1}$: $pValue = 6.47129e-07$

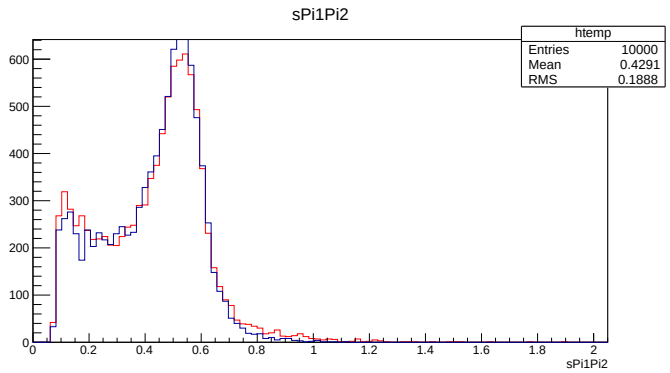


Figure 18. $s_{\pi_1\pi_2}$: $pValue = 0.00755842$

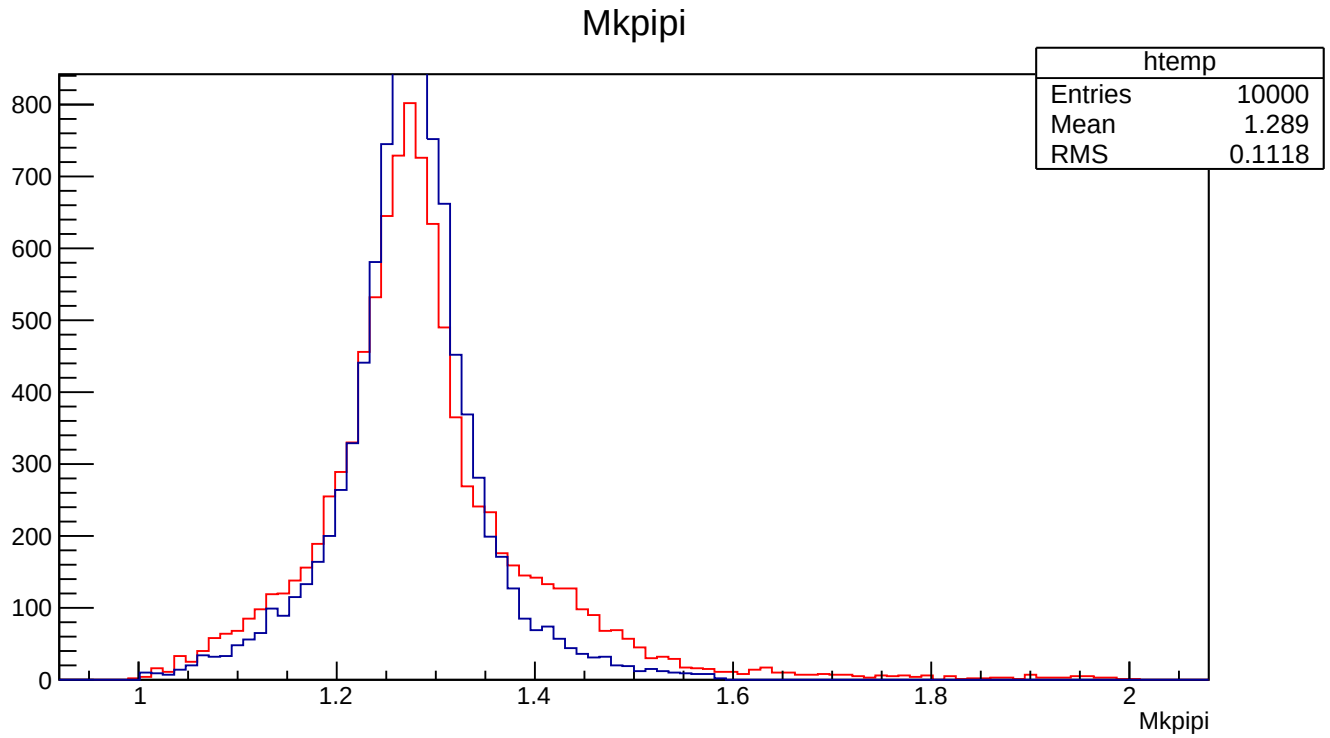


Figure 19. $M_{K\pi\pi}$: $pValue = 5.73461e-28$

```
1 thetaK1        :    truth:    1.042     fit:      2.046e-01    error:     5.260e-03
2 phiDK*         :    truth:    0.000     fit:      4.224e+00    error:     3.919e-02
3 phiSRho        :    truth:    0.000     fit:      3.651e+00    error:     2.781e-02
4 phiDRho        :    truth:    0.000     fit:      6.060e+00    error:     1.408e-01
5 lambda         :    truth:    0.700     fit:      5.124e-01    error:     3.631e-02
```

Listing 11. Table of fit parameters using K1(1270) model

For two cases with full and reduced models, distributions of angular projections look similar. Since K_1^{1270} is dominant corresponding angular distributions are very close and pValues are uniformly distributed. Meanwhile $M_{K\pi\pi}$ distributions are different and abscence of $1400, 1410, 1430$ is noticable.

3.3. Sensitivity studies by events number

All fits are made within: $1 < M_{K\pi\pi} < 1.6\ GeV$. For each new fit new sample of events was generated. Seed for the generator is determined from internal clock.

- 1000 events:

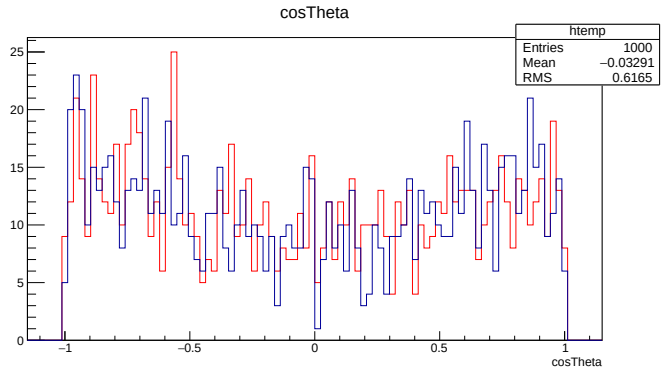


Figure 20. $\cos\theta$: $pValue = 0.0930233$

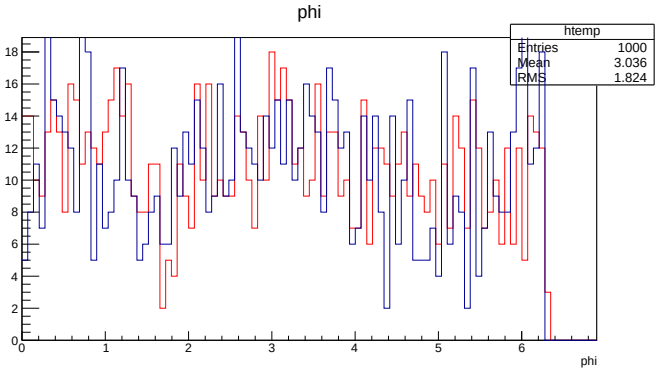


Figure 21. φ: $pValue = 0.882892$

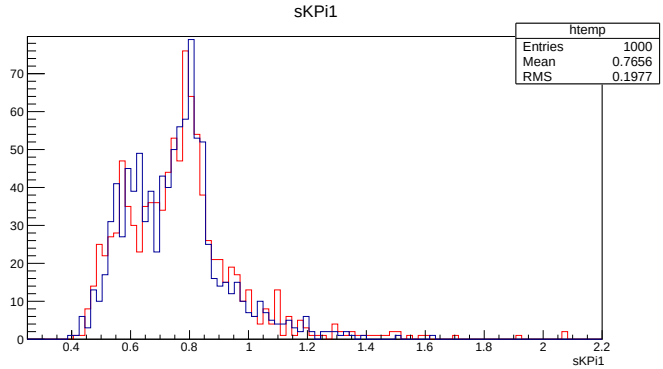


Figure 22. $s_{K\pi_1}$: $pValue = 0.332679$

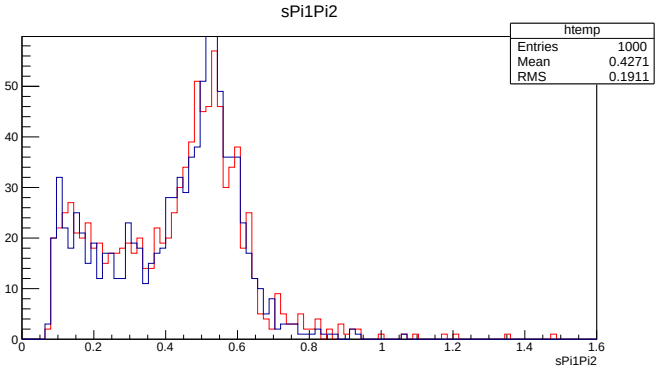


Figure 23. $s_{\pi_1\pi_2}$: $pValue = 0.638754$

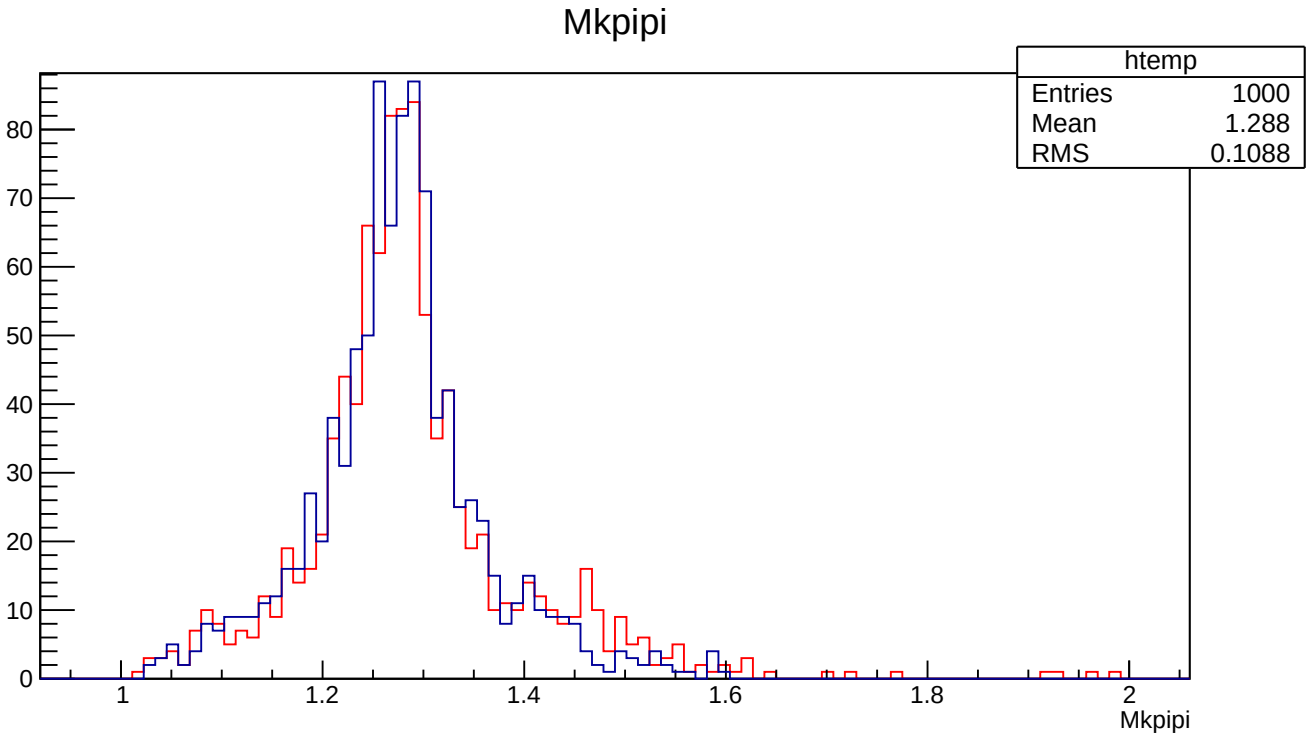


Figure 24. $M_{K\pi\pi}$: $pValue = 0.361409$

```
thetaK1        :    truth:    1.0420000000000000e+00     fit:      3.4133694434354291e+00    error:     2.3602260874495688e-02
phiDK*         :    truth:    0.0000000000000000e+00     fit:      3.2080736271984418e+00    error:     2.5554718039278673e-01
phiSRho        :    truth:    0.0000000000000000e+00     fit:      2.9711277697355727e+00    error:     1.0376189649267366e-01
phiDRho        :    truth:    0.0000000000000000e+00     fit:      8.8012765425722765e+00    error:     4.2432315484811456e-01
lambda         :    truth:    6.9999999999999996e-01     fit:      7.6237017570354004e-01    error:     1.0283057358491997e-01
ffRe_1400      :    truth:    4.6999999999999997e-01     fit:     -3.9364909505995987e-01     error:     4.1661495828085565e-02
ffIm_1400      :    truth:    0.0000000000000000e+00     fit:      3.4896193785163711e-02    error:     5.1475611573645080e-02
gg3Kstr_1410   :    truth:    1.0000000000000000e+00     fit:      1.1490088579418734e+00    error:     7.9925240513450080e-01
gg3ImKstr_1410 :    truth:    0.0000000000000000e+00     fit:      7.8392159399859951e-01    error:     7.8550576174695841e-01
gg3Rho_1410    :    truth:    1.0000000000000000e+00     fit:     -1.5379453649757926e+00     error:     1.4072959666495910e+00
gg3ImRho_1410  :    truth:    0.0000000000000000e+00     fit:      1.4764997643128291e-01    error:     1.4725232720497579e+00
gg4Kstr_1680   :    truth:    1.0000000000000000e+00     fit:      2.8703828735452347e-01    error:     5.7883129273077272e-01
gg4ImKstr_1680 :    truth:    0.0000000000000000e+00     fit:     -6.3655622858674166e-01     error:     5.7342039835072611e-01
gg4Rho_1680    :    truth:    1.0000000000000000e+00     fit:      2.3330902627954159e-01    error:     8.6696417095975986e-01
gg4ImRho_1680  :    truth:    0.0000000000000000e+00     fit:      3.3328440418606009e-01    error:     9.0219184142599174e-01
gg5Kstr_1430   :    truth:    1.0000000000000000e+00     fit:      7.3402828506452855e-01    error:     9.0400770815099715e-02
gg5ImKstr_1430 :    truth:    0.0000000000000000e+00     fit:      1.6402405811993131e-02    error:     1.3246104681228482e-01
gg5Rho_1430    :    truth:    1.0000000000000000e+00     fit:      3.3442494103289760e-01    error:     1.8094855039882443e-01
gg5ImRho_1430  :    truth:    0.0000000000000000e+00     fit:     -9.6277476425554037e-02     error:     1.9187336731017662e-01
```

Figure 25. Fit table with model parameters for 1000 events

- 5000 events:

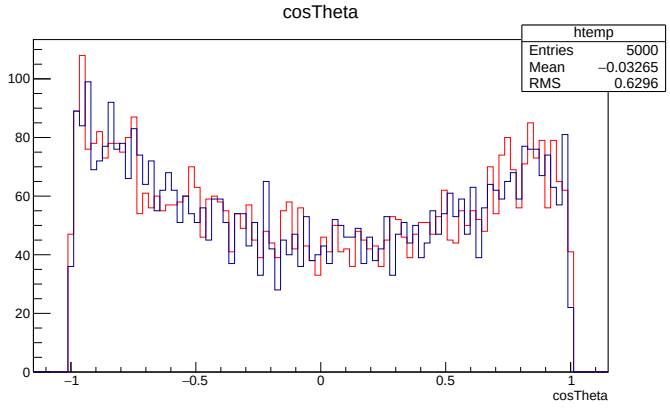


Figure 26. $\cos\theta$: $pValue = 0.973352$

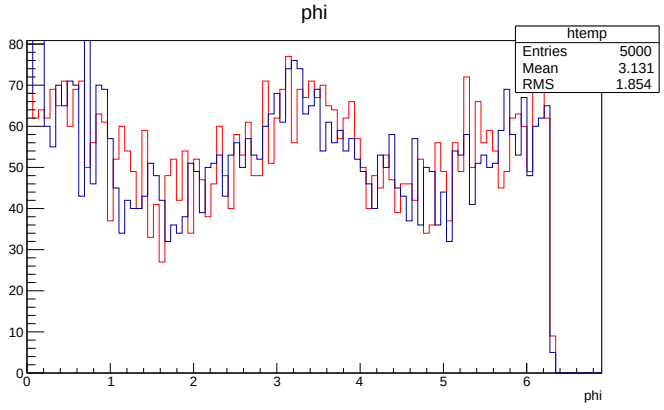


Figure 27. φ: $pValue = 0.425322$

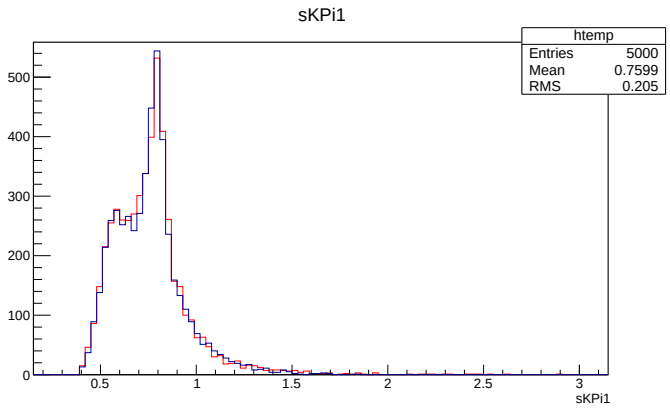


Figure 28. $s_{K\pi_1}$: $pValue = 0.583377$

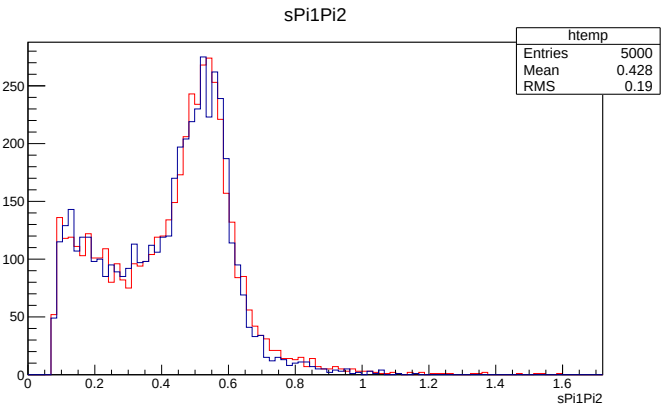


Figure 29. $s_{\pi_1\pi_2}$: $pValue = 0.455054$

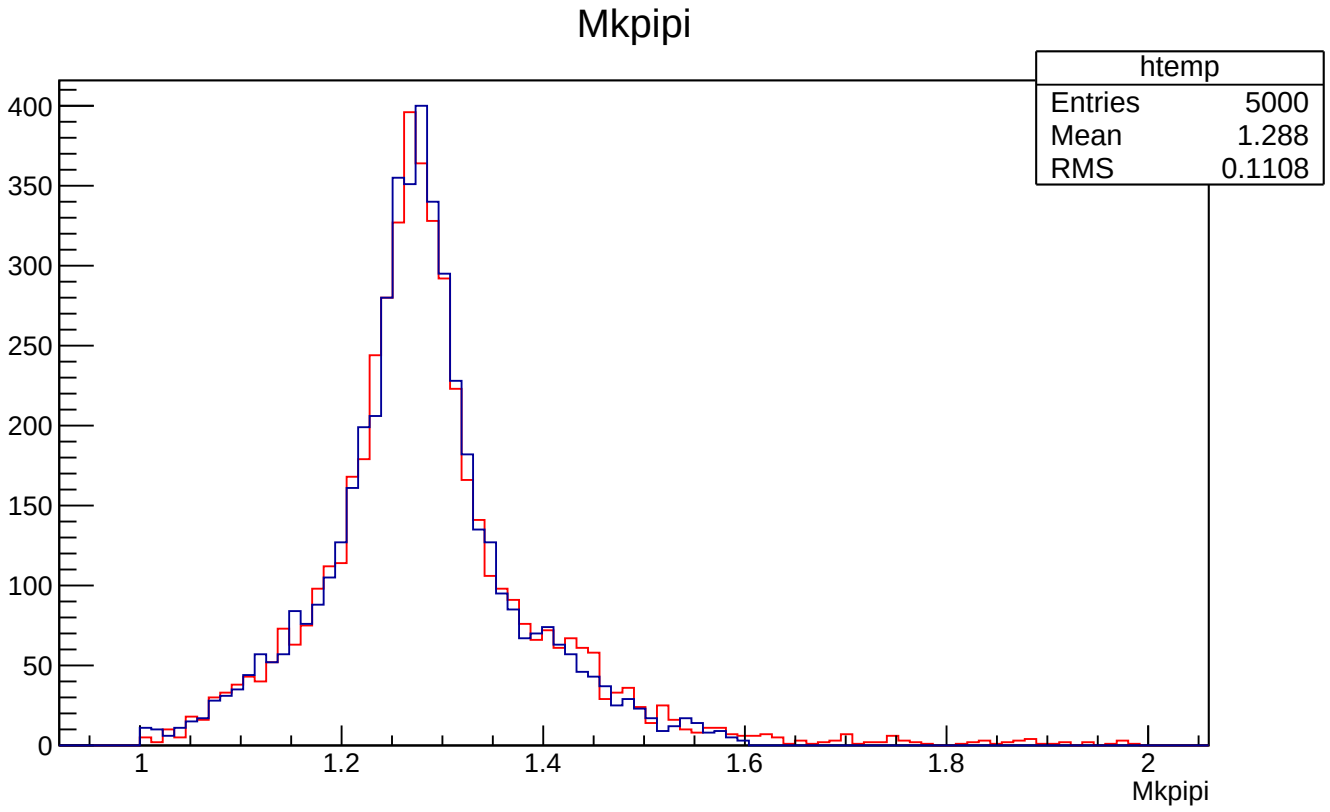


Figure 30. $M_{K\pi\pi}$: $pValue = 0.470322$

thetaK1	: truth:	1.0420000000000000e+00	fit:	1.0472601451328025e+00	error:	1.3966399709773232e-02
phiDK*	: truth:	0.0000000000000000e+00	fit:	6.2898438879996679e+00	error:	9.2286715900266200e-02
phiSRho	: truth:	0.0000000000000000e+00	fit:	2.8688788347067339e-03	error:	4.8419340085988957e-02
phiDRho	: truth:	0.0000000000000000e+00	fit:	1.0073581949054176e-01	error:	3.5669482103000844e-01
lambda	: truth:	6.9999999999999996e-01	fit:	6.9993570534704042e-01	error:	5.2018204672717519e-02
ffRe_1400	: truth:	4.6999999999999997e-01	fit:	4.8242871531435205e-01	error:	2.4663653231465171e-02
ffIm_1400	: truth:	0.0000000000000000e+00	fit:	4.9601587059992297e-03	error:	3.4119140867018491e-02
gg3Kstr_1410	: truth:	1.0000000000000000e+00	fit:	2.2482399131215716e+00	error:	4.1653964670004201e-01
gg3ImKstr_1410	: truth:	0.0000000000000000e+00	fit:	3.9960927900863408e-01	error:	4.8861473534315952e-01
gg3Rho_1410	: truth:	1.0000000000000000e+00	fit:	8.3751499483921277e-01	error:	8.0523648463970676e-01
gg3ImRho_1410	: truth:	0.0000000000000000e+00	fit:	-6.1712681972310668e-01	error:	7.9150491256910649e-01
gg4Kstr_1680	: truth:	1.0000000000000000e+00	fit:	-1.9511099808898122e-01	error:	3.2355170812676715e-01
gg4ImKstr_1680	: truth:	0.0000000000000000e+00	fit:	-5.5239119326942221e-01	error:	3.4311490358412644e-01
gg4Rho_1680	: truth:	1.0000000000000000e+00	fit:	-4.4557334749492411e-01	error:	4.8995549077883926e-01
gg4ImRho_1680	: truth:	0.0000000000000000e+00	fit:	8.5595104341540107e-02	error:	5.0266776536403734e-01
gg5Kstr_1430	: truth:	1.0000000000000000e+00	fit:	8.4466178341785525e-01	error:	5.9499936934204779e-02
gg5ImKstr_1430	: truth:	0.0000000000000000e+00	fit:	1.9298647974029426e-01	error:	8.4723025149911854e-02
gg5Rho_1430	: truth:	1.0000000000000000e+00	fit:	7.7149924101614231e-01	error:	1.0008519020741252e-01
gg5ImRho_1430	: truth:	0.0000000000000000e+00	fit:	1.0771415860810744e-01	error:	1.1569720592284467e-01

Figure 31. Fit table with model parameters for 5000 events

- 20000 events:

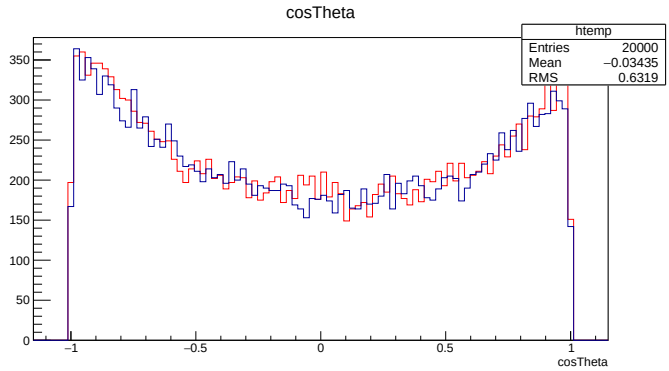


Figure 32. $\cos\theta$: $pValue = 0.70109$

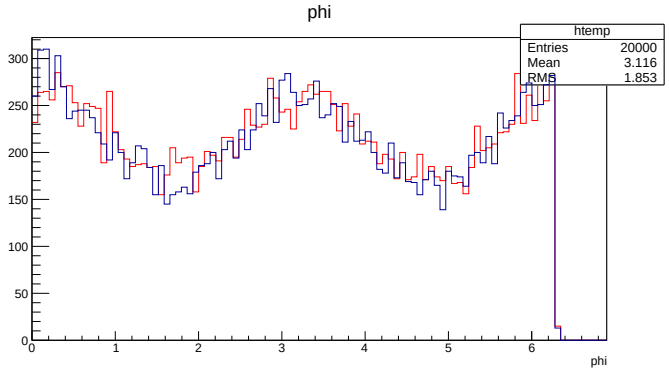


Figure 33. φ: $pValue = 0.355259$

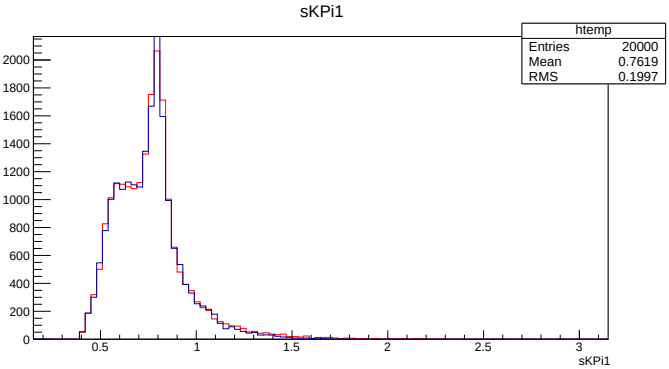


Figure 34. $s_{K\pi_1}$: $pValue = 0.775272$

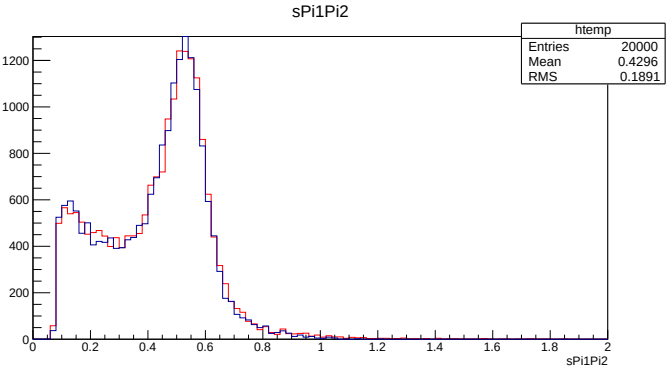


Figure 35. $s_{\pi_1\pi_2}$: $pValue = 0.410014$

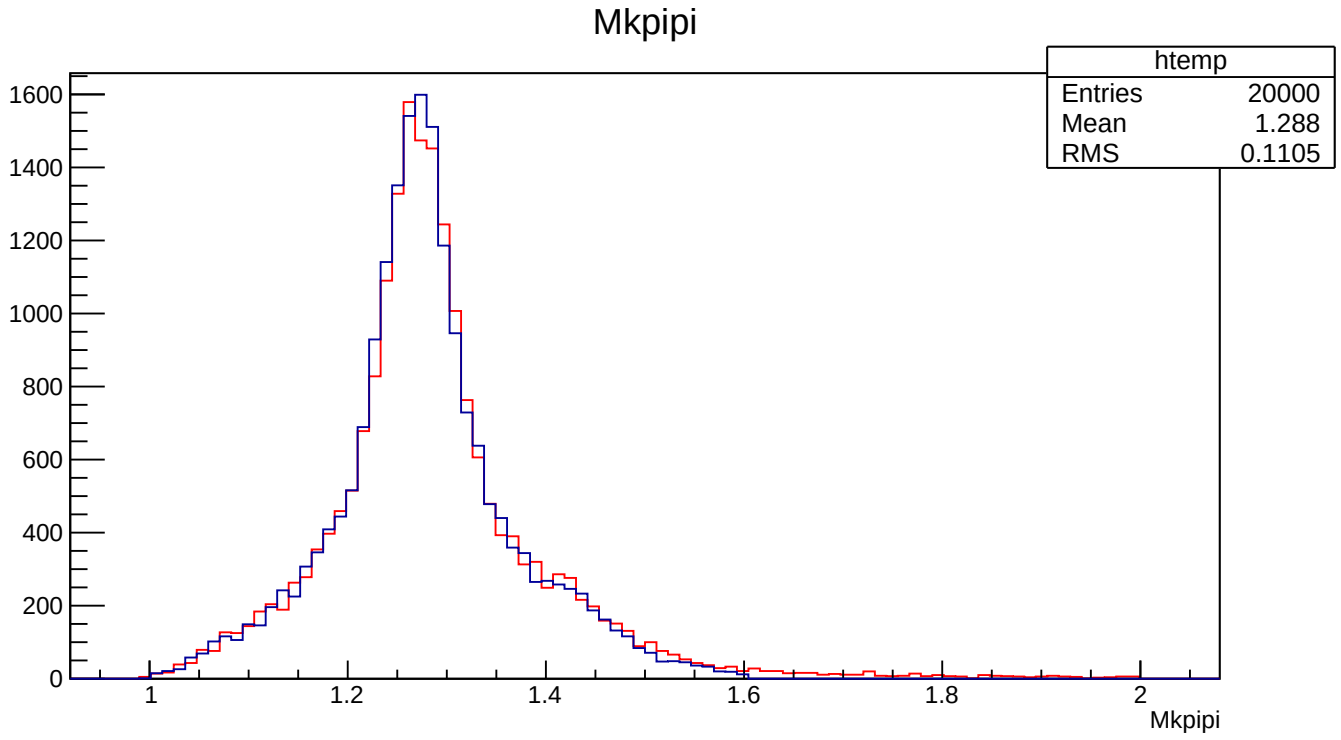


Figure 36. $M_{K\pi\pi}$: $pValue = 0.00640837$

thetaK1	:	truth:	1.0420000000000000e+00	fit:	1.0387890910378199e+00	error:	6.9222802670752762e-03
phiDK*	:	truth:	0.0000000000000000e+00	fit:	6.1952496783141138e+00	error:	4.9638938238082402e-02
phiSRho	:	truth:	0.0000000000000000e+00	fit:	-8.9730944540036377e-03	error:	2.3928941528120196e-02
phiDRho	:	truth:	0.0000000000000000e+00	fit:	-1.2057416900939512e-01	error:	1.6857415875766632e-01
lambda	:	truth:	6.9999999999999996e-01	fit:	6.8350008034509735e-01	error:	2.5166557578049944e-02
ffRe_1400	:	truth:	4.6999999999999997e-01	fit:	4.6856727573663653e-01	error:	1.1338762540206375e-02
ffIm_1400	:	truth:	0.0000000000000000e+00	fit:	-2.9688173016207502e-02	error:	1.6656688503338905e-02
gg3Kstr_1410	:	truth:	1.0000000000000000e+00	fit:	1.2002598749252931e+00	error:	1.9456040081684492e-01
gg3ImKstr_1410	:	truth:	0.0000000000000000e+00	fit:	-6.2561660611029091e-01	error:	2.4630333836772628e-01
gg3Rho_1410	:	truth:	1.0000000000000000e+00	fit:	8.8728173760056772e-01	error:	3.7467609455727918e-01
gg3ImRho_1410	:	truth:	0.0000000000000000e+00	fit:	-6.6410322028655533e-01	error:	4.0413485576866820e-01
gg4Kstr_1680	:	truth:	1.0000000000000000e+00	fit:	-2.6927047661892461e-01	error:	1.6310275153668746e-01
gg4ImKstr_1680	:	truth:	0.0000000000000000e+00	fit:	-1.7786940599054818e-01	error:	1.5378073764787825e-01
gg4Rho_1680	:	truth:	1.0000000000000000e+00	fit:	-2.6933662782879941e-01	error:	2.4599179259984380e-01
gg4ImRho_1680	:	truth:	0.0000000000000000e+00	fit:	-1.1828642592774426e-01	error:	2.3686229586695534e-01
gg5Kstr_1430	:	truth:	1.0000000000000000e+00	fit:	8.2140069751011957e-01	error:	2.6313130339904945e-02
gg5ImKstr_1430	:	truth:	0.0000000000000000e+00	fit:	-8.2410965414454283e-03	error:	3.9332104412693289e-02
gg5Rho_1430	:	truth:	1.0000000000000000e+00	fit:	8.7484802721226740e-01	error:	4.7632221112326630e-02
gg5ImRho_1430	:	truth:	0.0000000000000000e+00	fit:	-7.3801332513158555e-02	error:	5.9691943341244856e-02

Figure 37. Fit table with model parameters for 20000 events

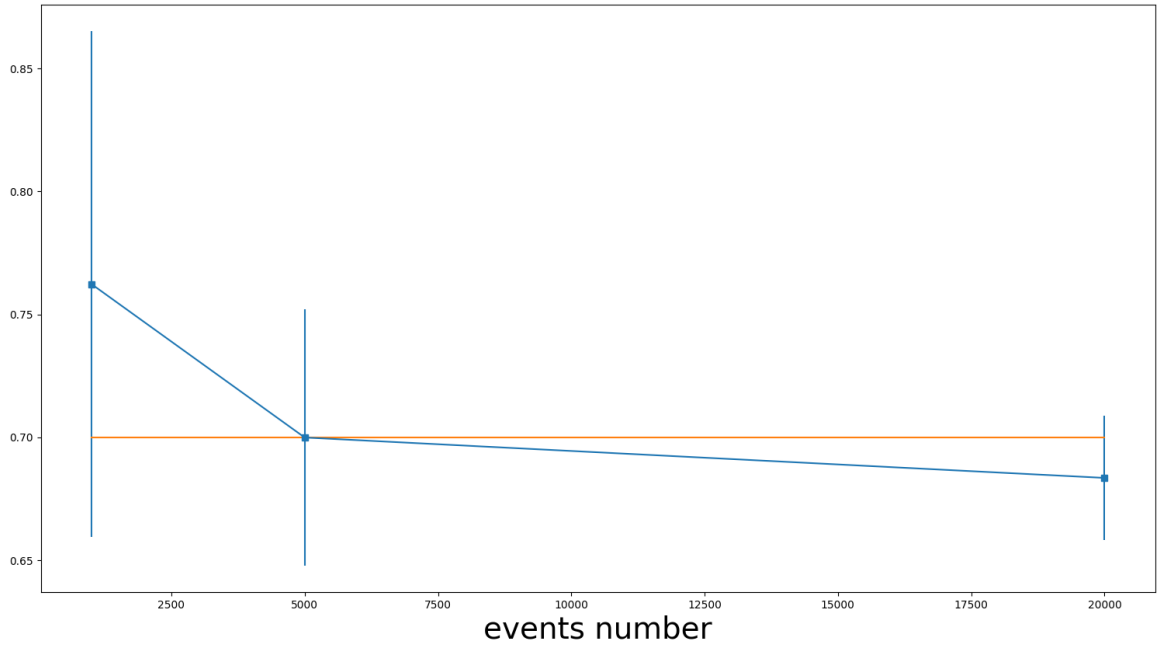


Figure 38. Fit table with model parameters for 1000 events

One can note that uncertainty scales down with $\sqrt{N_{evts}}$. Fitting of K^{*1680} is difficult since it is out of range $1 < M_{K\pi\pi} < 1.6\ GeV$.

3.4. Sensitivity studies by $M_{K\pi\pi}$ cut

The purpose of this study is to understand how tight could be $M_{K\pi\pi}$ cut. This cut is useful to suppress background and save sensitivity to the measurements of λ.

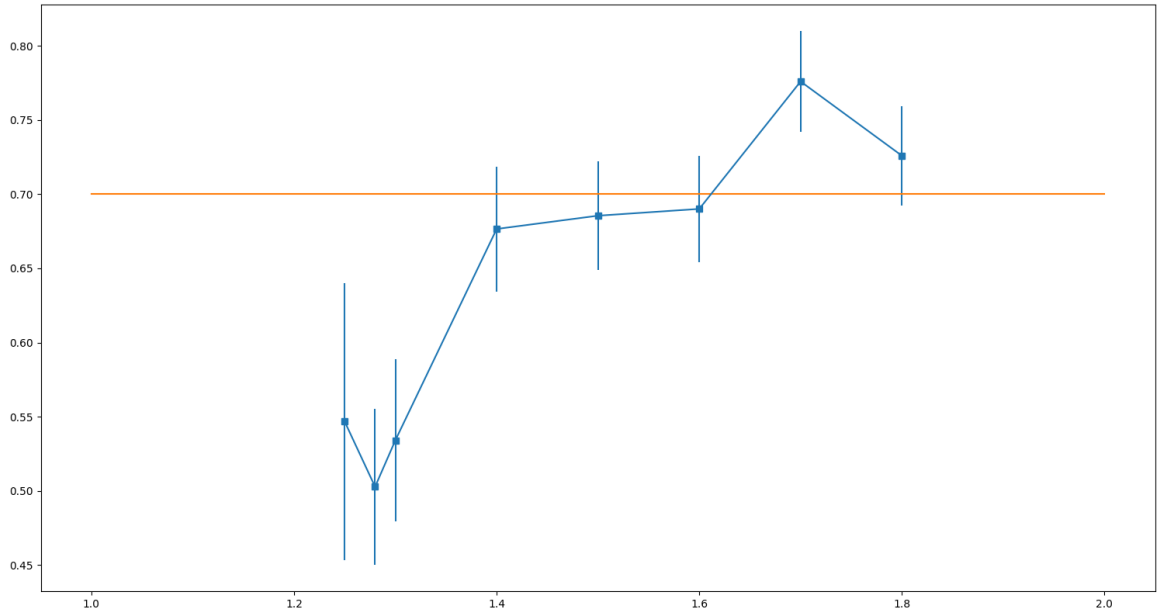


Figure 39. Sensitivity studies by cut

This preliminary study shows that one can apply cut $1 < M_{K\pi\pi} < 1.4\ GeV$ and measure photon polarization parameter.

4. Symbolic computations

This section describes details of symbolic computation approach used for calculation of noramlization integrals. Computations rely on C++ framework which is called "GiNaC". Kinematical numerical coefficients (K_t^{num}) are calculated implicitly on caching step inside numerical decay rate such that:

$$F^{num}(\overrightarrow{\Omega}, \overrightarrow{\alpha}^{num}) = K_1^{2,num}\alpha_1^{2,num} + \alpha_{2,num}K_{10}^{num}(K_2^{num} + \alpha_1^{2,num}K_4^{num}) + ... + \alpha_{40}^{2,num}K_{39}^{num} + ... \tag{122}$$

Also kinematical symbolic coefficients (K_t^{sym}) are defined in symbolic computations module:

$$F^{sym}(\overrightarrow{\Omega}, \overrightarrow{\alpha}^{sym}) = K_1^{2,sym}\alpha_1^{2,sym} + \alpha_{2,sym}K_{10}^{sym}(K_2^{sym} + \alpha_1^{2,sym}K_4^{sym}) + ... + \alpha_{40}^{2,sym}K_{39}^{sym} + ... \tag{123}$$

The point here to note that structure of symbolic and numerical decay rate functions should have the same structure.[5] This is unexpanded representation of decay rate function, because it doesn't follow the law defined in factorization expression Eq. 115. K_t^{num} are building blocks of each $f_i(\overrightarrow{\Omega})$ function as following:

$$f_i(\overrightarrow{\Omega}) = \prod_{t=0}^{N_i} K_t^{num} \tag{124}$$

Even though kinematical coefficients are calculated numerically, there is no information provided which exactly kinematical coefficients should couple together in order to produce some particular f_i. One needs a tool which solves the following task: *Given kinematical coefficients from Eqn. 124, split decay rate in a way as in Eqn. 115 and assign for each term product of given kinematical coefficients.* This is how the task is solved on the **caching** stage:

- initialization by providing the same information as to the *GMathFunction* and creating symbolic variables for model parameters;
- delete directory with previous normalization integrals calculation;
- create new directory for writing there newly created integrals;

[5] Such "double" implementation of the same function could seem misleading on the first look. One can say, that why not to define symbolic decay rate once, compile and use in the same way as for normalization integrals calculation?

- construct symbolic analogue of numerical decay rate and fill K^{symb} list of symbolic kinematical variables. The order of coefficients corresponds to the order of coefficients in the K^{num};
- split symbolical pdf on two parts (see Eqn.2): *kinematical terms* and *model parameters terms*;
- write each of these parts (*archivate*) to separate *.gar* files. This is useful when structure of decay rate is changed by introduced new kinematical terms or added new couplings.;
- *unarchivate .gar* files. If some K_t^{num} are changed for some reasons it doesn't affect symbolic decay rate structure, but affects values of $f_i(\overrightarrow{\Omega}_j)$. Hence unarchivation is required to restore raw structure of symbolic decay rate;
- using kinematical terms and model parameters terms produce dynamic *.so* libraries (using *GExCompiler* class).
- for each event: *GMathFunctions* $\rightarrow K_t^{num} \rightarrow$ *kinematicsCoeffs.so* $\rightarrow f_i(\overrightarrow{\Omega}_j)$;
- $\tilde{f}_i = \left[\sum_{j=0}^{N_\Sigma} f_i(\overrightarrow{\Omega}_j)\right]$ — normalization integrals calculation (*GEventsAnalyzer* class);
- writing $\tilde{f}_i$ in the *.txt* file;

Fitting stage:

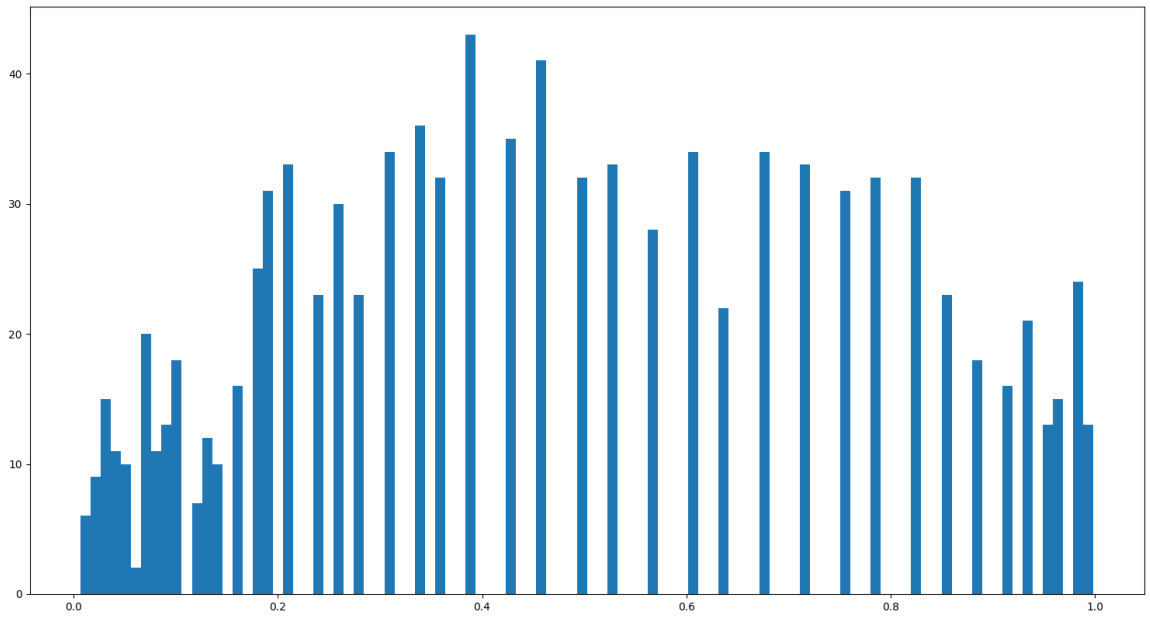
- read normalization integrals from *.txt* file;
- read *.root* file in order to extract data;
- get poiter to function from *couplingCoeffs.cxx.so* library;
- *.txt, .root, .so, .fit* $\rightarrow$*objective function* (see Eqn.8) $\rightarrow$ *GEventsAnalyzer* $\rightarrow$ *ROOT::MinuitMinimizer* $\rightarrow$ result;

5. Appendix

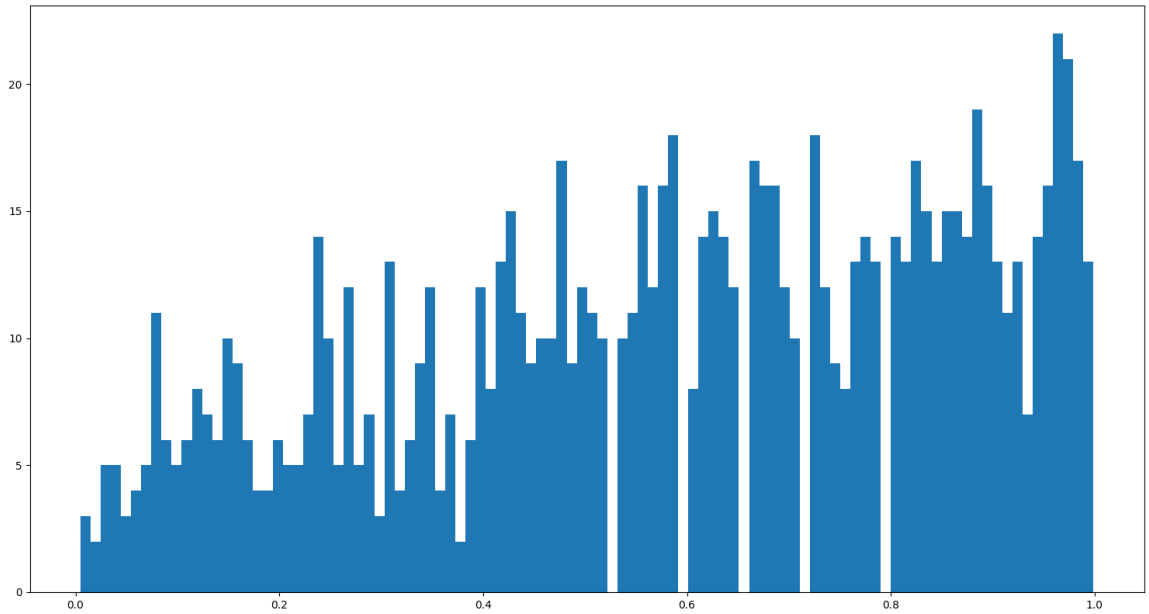
5.1. p-Value distributions

Distributions of pValues obtained from KS-test using gaussian distributions.

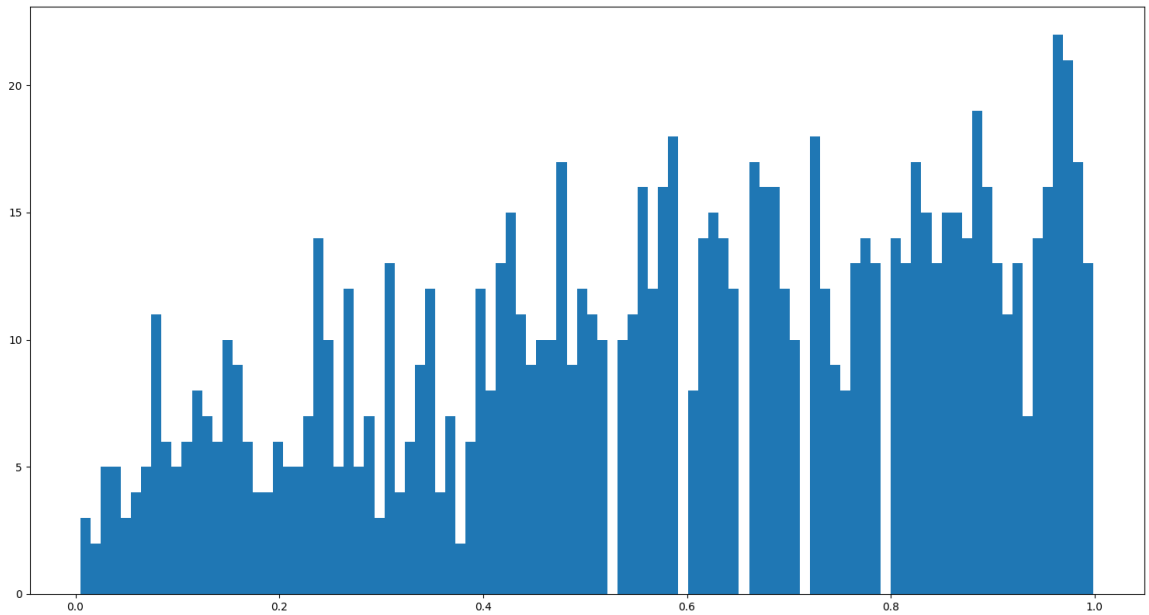
- 1000 vs 1000:



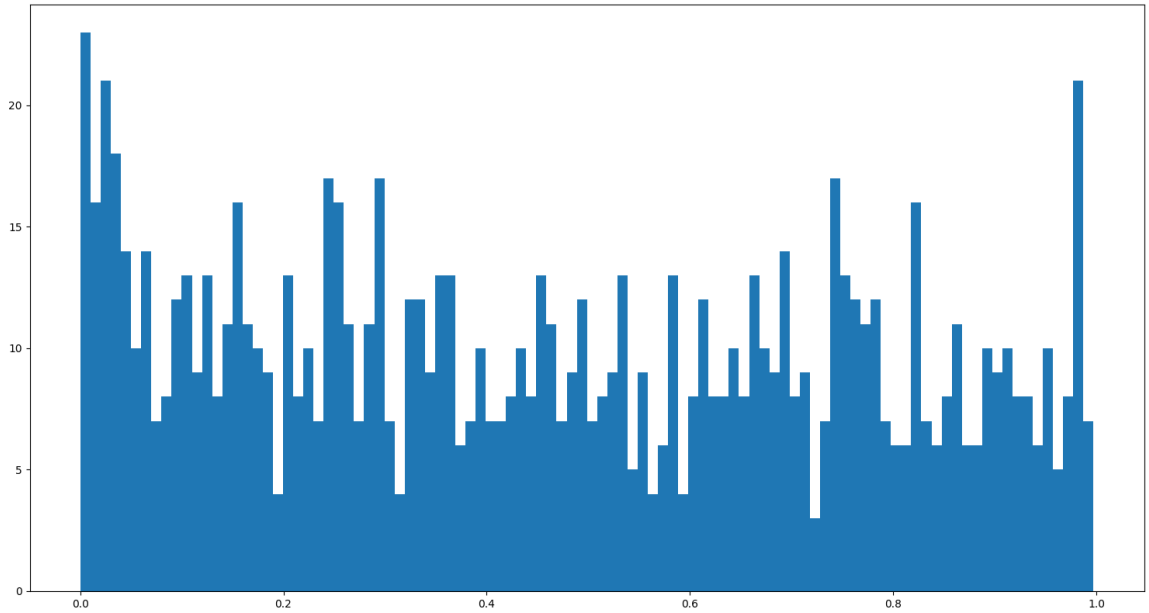
- 10000 vs 10000:



- 100000 vs 1000:



- 100000 vs 10000:



- 100000 vs 100000: