Track Reconstruction in KM3NeT

Brían Ó Fearraigh

June 29, 2020

1 Intro

This is - at the moment - meant to be a brief introduction to the reconstruction process in KM3NeT, and what it all means.
This document is meant to be a guide for newcomers, but also contain information useful to those familiar with the code.

2 What Is Reconstruction?

At the end of the day, reconstruction means fitting a model (specifically, model parameters) to data. The underlying, most important one for us is the Cherenkov cone hypothesis. Particles with enough energy emit Cherenkov radiation, which propogates through our detector as a cone of light. Neutrino interactions are usually separated into 'track' events and 'shower' events. A neutrino interaction, depending on the neutrino flavour, can result in a muon coming out of the interaction point - a track - or can create a cascade of particles - hadrons or electrons - which is termed a shower. The separate event types are dealt with using separate reconstruction algorithms. The model we use assumes that a muon traverses the detector and emits Cherenkov radiation as it propogates, and the same for particles produced in a shower.

Now, what do we want to 'reconstruct'? We settle for wanting to find out the energy and direction of the muon tracks, as well as the energy and direction of showers. In muon track reconstruction, the track length is very much tied to the energy of the muon. Approximately, a 1 GeV muon travels 4 metres. For showers, this is trickier. A shower profile is not simply proportional to its energy: the number of particles varies, for example. ORCA and ARCA employ different shower reconstruction algorithms, because of the geometrical difference between the detectors. For **track reconstruction** the method is practically detector-independent.
This act of reconstructing tracks and showers is not so simple. In KM3NeT there is a 'chain' of steps taken to carry out track reconstruction. If someone refers to the '*reco chain*' or '*JChain*', this is what they mean. The chain looks like this:

JPrefit $\longrightarrow$ JGandalf $\longrightarrow$ JStart $\longrightarrow$ JEnergy.

Now, a few brief notes on this chain. There used to be a 'JSimplex' between JPrefit and JGandalf, but this is no longer included in the reconstruction chain. This step was removed after a conclusion that JSimplex does not contribute much to the angular resolution (to what precision we can reconstruct the muon direction); the study can be found on the eLog here
New Optimisation for the ORCA JGandalf Chain .

The input to this chain is an event file. Usually, you want to take an event file that has been processed by JTriggerEfficiency. JTriggerEfficiency simulates the background light (from K-40 decays), simulates the response of PMTs to photon hits, and applies a trigger to these hits. The neutrino events in these files are termed 'post-trigger'. Also, take note that the order of this chain matters, and JEvt is applied to the files that come after JEnergy. This converts the .root formatted files to aanet.root format.

3 Track Reconstruction

The following section aims to give a brief, not overly technical description of these steps in the reconstruction chain. If more information is needed, please consult the code itself.
We want to fit to our data the hypothesis that a muon track causes hits in the KM3NeT detector. In reconstructing a muon event, we want to describe the important parameters: the position and direction of the muon at each point in time, giving 5 independent parameters. This is a non-linear problem.
In this case, performing an inital fit gives us a starting point for determining the muon direction. Neglecting light scattering and assuming a muon direction reduces the non-linear problem to a linear one. Then a full fit and maximum likelihood estimation can be carried out, followed by an estimation of the muon energy.
If you are not too familiar with *JPP*, the script name followed by -h! prints out a small description of the code and the list of default values that are used. Note that the 'default' parameters when running this chain differ between ARCA and ORCA. The text files with the reference values an be found in *JPP_DATA*[1].

3.1 JPrefit

JPrefit does what it says on the tin. It performs an initial fit in the detector, taking N track direction hypotheses in all directions. By assuming a direction, the remaining parameters to fit to - or 'reconstruct' - are the position of the muon and time at which it crosses some reference plane perpendicular to the muon direction. The parameters describing the muon are obtained by an iterative procedure to minimise the χ^2. The χ^2 is based on the time difference between the photon arrival time and its expected arrival time, and is normalised to an assumed time resolution.
JPrefit saves a specified amount of these fits, and then hands them to JGandalf to perform the full direction fit and reconstruct the muon trajectory.

A coordinate system is defined by taking an assumed muon direction, where the muon travels parallel to the z-axis and crosses the $z = 0$ plane at (x_0,y_0) at time t_0. The muon trajectory passes the PMT of the detector at a *distance of closest approach* R, defined by

$$R_j = \sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}\,. \tag{1}$$

The expected arrival time t_j of the Cherenkov photons on the PMT j can be expressed as

$$t_j = t_0 + \frac{z_j}{c} + \tan(\theta_c)\frac{R_j}{c}\,, \tag{2}$$

with z_j the distance to the z=0 plane, c the speed of light in a vacuum, and θ_c the Cherenkov angle. Defining:

[1]These values are configurable, and investigations into changing the input parameters for JPrefit $\rightarrow$ JGandalf are described in the eLog here and here . Note the roadwidth is an important parameter; JPrefit and JGandalf appear to need a roadwidth of $\approx$ 50 metres for a clean hit selection.

$$t'_j = t_j c/\tan(\theta_c) - z_j/\tan(\theta_c) \tag{3}$$

and

$$t'_0 = t_0 c/\tan(\theta_c), \tag{4}$$

this gives, for all pairs of consecutive hits i,j the relation:

$$\begin{aligned} t'^2_j - t'^2_i - 2(t'_j - t'_j)t'_0 = x_j^2 - x_i^2 - 2(x_j - x_j)x_0 \\ + y_j^2 - y_i^2 - 2(y_j - y_j)y_0 \,. \end{aligned} \tag{5}$$

See A.1 for the derivation.
Note that x_0, y_0 and t_0 appear in a linear manner. This is solved in a matrix equation by considering all pairs of consecutive hits, which amounts to the number of hits n. See A.2 for further details.
In JPrefit, a cluster of causally related hits is selected from the data (in 1D), in order to minimise the linear fit being affected by the optical background. Possible outliers are removed as long as their contribution to the total χ^2 is larger than 3 standard deviations.
This procedure is repeated for a specified grid angle (the default value is currently 5° for ORCA), which is used to scan across the whole solid angle of the sky. The N best-fit test directions are stored and used in the next stage of the reconstruction as start points for a full fit. The 'best' fit is determined by the fit quality parameter Q (something that appear often), and in JPrefit is given by

$$Q = \mathrm{NDF} - \frac{1}{4}\frac{\chi^2}{\mathrm{NDF}}\,, \tag{6}$$

where NDF is the number of degrees of freedom in the fit.

3.2 JGandalf

JGandalf is our be-all and end-all of track direction reconstruction. Taking the fits from JPrefit (either all the fits or the fit with the highest quality, you decide), JGandalf takes these starting points, performs a scan *around* these fits and carries out the full fit. Again, 3D clustering of causally related hits takes place.
This full fit (to all five parameters) incorporates the Levenburg-Marquardt method for the likelihood scan. This likelihood uses a set of PDFs to describe the PMT response:

$$\mathcal{L} = \prod_{\text{hit PMTs}} \left[\frac{\delta P}{\delta t}\,(R_i, \theta_i, \phi_i, \Delta t)\right], \tag{7}$$

where R_i is the distance of closest approach (of the muon the the ith PMT) as previous, θ and ϕ describe the orientation of the PMT and Δt is the difference between the expected and measured arrival time of light (see Equation 2). The set of PDFs are calculated semi-analytically for direct and single-scattered light from Cherenkov radiation and from energy losses of the muon, such as Bremsstrahlung. These PDFs also take into account the effects of the optical background and dispersion of light, as well as the quantum efficiency, angular acceptance, and transit time spread of the PMTs. The PDFs are evaluated using interpolating functions in 4D. Currently, JGandalf does not include the direct or scattered light due to delta-rays (secondary ionised particles) in the PDFs.
A good performance check of JGandalf is the resulting angular deviation (the angle which comes from the dot product of the reconstructed and true direction), and also the median angular resolution, which can be found from the spread of the angular deviation as a function of the muon/neutrino energy.
JGandalf also has a quality parameter associated with it. In this case, the quality $Q = -\chi^2$. The larger the quality, the better the fit.

3.3 JStart

In JStart, the observed start position of muon trajectory is determined by back-projecting the hits onto the track under the Cherenkov angle. The first associated emission point which exceeds the random background level is selected as the start position.

3.4 JEnergy

JEnergy determines the energy. It should be ran after JStart, because knowing the track length is useful for finding the energy!
The energy is fitted using the spatial distribution of hit and non-hit PMTs. All PMTs within a pre-defined roadwidth around the muon trajectory are used, for which the calculated hit probability for a PMT is compared to the occurence of a hit. The procedure involves a 1 parameter likelihood fit of $\log E$.

References

[1] ANTARES-SOFT-2007-001, M. de Jong.

[2] KM3NeT/ARCA Event Reconstruction Algorithms, Melis et al. 2017.

A Appendix

A.1 Derivation of Hit Relation

From Equation 3 and 4 we get the expressions

$$t_j = t'_j \tan(\theta_c) + z_j/c \tag{8}$$

and

$$t_0 = t'_0 \tan(\theta_c)/c\,. \tag{9}$$

Substituting these into Equation 2 gives:

$$t'_j \tan(\theta_c)/c + z_j/c = t'_0 \tan(\theta_c)/c + z_j/c + \tan(\theta_c)\frac{R_j}{c} \tag{10}$$

$$\implies t'_j = t'_0\, R_j\,. \tag{11}$$

So,

$$t'_j - t'_0 = R_j\,, \tag{12}$$

$$\implies t'_j - t'_0 = \sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}\,, \tag{13}$$

$$\implies \left(t'_j - t'_0\right)^2 = (x_j - x_0)^2 + (y_j - y_0)^2\,. \tag{14}$$

This equation corresponds to a cone and relates the **fit parameters** (x_0, y_0, t'_0) to the **hit parameters** (x_j, y_j, t'_j)

$$\begin{aligned} \therefore\ t'^2_j - 2t'_j t'_0 + t'^2_0 = x_j^2 - 2x_j x_0 + x_0^2 \\ +y_j^2 - 2y_j y_0 + y_0^2\,. \end{aligned} \tag{15}$$

Following the same logic for a hit on the PMT i with expected arrival time t_i (with distance of closest approach R_i)

$$\begin{aligned} t'^2_i - 2t'_i t'_0 + t'^2_0 = x_i^2 - 2x_i x_0 + x_0^2 \\ +y_i^2 - 2y_i y_0 + y_0^2\,. \end{aligned} \tag{16}$$

Subtracting Equation 16 from Equation 15:

$$\begin{aligned} t'^2_j - t'^2_i - 2t'_j t'_0 + 2t'_i t'_0 = x_j^2 - x_i^2 - 2x_j x_0 + 2x_i x_0 \\ +y_j^2 - y_i^2 - 2y_j x_0 + 2y_i x_0\,. \end{aligned} \tag{17}$$

This gives the result shown in Equation 5

$$\begin{aligned} t'^2_j - t'^2_i - 2(t'_j - t'_j)t'_0 = x_j^2 - x_i^2 - 2(x_j - x_j)x_0 \\ + y_j^2 - y_i^2 - 2(y_j - y_j)y_0\,. \end{aligned} \tag{18}$$

A.2 Matrix Equation

Equation 5 can be formulated in a matrix equation considering all pairs of consecutive hits:

$$H\,\Theta = Y \tag{19}$$

where

$$H = \begin{pmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) & -2(t'_2 - t'_1) \\ 2(x_3 - x_2) & 2(y_3 - y_2) & -2(t'_3 - t'_2) \\ . & . & . \\ . & . & . \\ 2(x_1 - x_n) & 2(y_1 - y_n) & -2(t'_1 - t'_n) \end{pmatrix}, \tag{20}$$

$$Y = \begin{pmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 - t_2'^2 + t_1'^2 \\ x_3^2 - x_2^2 + y_3^2 - y_2^2 - t_3'^2 + t_2'^2 \\ . \\ . \\ x_1^2 - x_n^2 + y_1^2 - y_n^2 - t_1'^2 + t_n'^2 \end{pmatrix}, \tag{21}$$

and

$$\Theta = \begin{pmatrix} x_0 \\ y_0 \\ t'_0 \end{pmatrix}. \tag{22}$$

The number of pairs of consecutive hits amounts to the number of hits n, taking into account the pair of the first and last hit. The optimal (least squares) solution to 19 is

$$\Theta = \left(H^T\,V^{-1}\,H\right)^{-1} \times H^T\,V^{-1} \times Y \tag{23}$$

with V the standard covariance matrix, which can be expressed as

$$V = J \times \begin{pmatrix} \sigma_1^2 & . & . & . & 0 \\ . & \sigma_2^2 & . & . & . \\ . & . & \sigma_3^2 & . & \\ . & . & . & . & . \\ 0 & . & . & . & \sigma_n^2 \end{pmatrix} \times J^T\,. \tag{24}$$

with σ_j the resolution of the measured time t_j.

The *Jacobian matrix* J is expressed as

$$J = \begin{pmatrix} 2t'_1 & -2t'_2 & 0 & . & . & . \\ 0 & 2t'_2 & -2t'_3 & 0 & . & . \\ . & 0 & 2t'_3 & -2t'_4 & . & . \\ . & . & . & . & . & . \\ -2t'_1 & . & . & . & . & 2t'_n \end{pmatrix}, \tag{25}$$

with $J_{ij} = \delta Y_i \ / \ \delta t_j$.

Thus the covariance matrix V is an $n \times n$ which at first order is (almost) tridiagonal:

$$V = \begin{pmatrix} (2t'_1\sigma'_1)^2 + (2t'_2\sigma'_2)^2 & -(2t'_2\sigma'_2)^2 & 0 & . & . & -(2t'_{21}\sigma_1)^2 \\ -(2t'_2\sigma'_2)^2 & (2t'_2\sigma'_2)^2 + (2t'_3\sigma'_3)^2 & -(2t'_3\sigma'_3)^2 & 0 & . & . \\ 0 & -(2t'_3\sigma'_3)^2 & (2t'_3\sigma'_3)^2 + (2t'_4\sigma'_4)^2 & . & . & . \\ . & 0 & . & . & . & . \\ . & . & . & . & . & . \\ -(2t'_1\sigma_1)^2 & . & . & . & . & (2t'_n\sigma'_n)^2 + (2t'_1\sigma'_1)^2 \end{pmatrix}, \tag{26}$$

where $\sigma'_j = \sigma_j c \ / \ \tan\theta_c$. Neglecting the off-diagonal elements of V, the number of operations needed to obtain the solution Θ is proportional to n (the number of hits).