Blind analysis in $\mathcal{TWIST}$

Technical note #88

Andrei Gaponenko

2003 December 30, revision 2

Abstract

This note describes an implementation of a blind analysis technique in $\mathcal{TWIST}$ and discusses some practical points of its use.

Introduction

The determination of Michel parameters in $\mathcal{TWIST}$ can be done by fitting a linear combination of histograms of reconstructed Monte-Carlo spectra to a histogram of reconstructed data spectrum, as described in [1]. In this approach all spectrum distortions caused by reconstruction software cancel between data and Monte-Carlo. Also, the resolution is automatically taken into account.

This method allows for a straightforward "blinding" of analysis: it is enough to hide the values of Michel parameters used to produce the reference Monte-Carlo spectrum [1], since the fit is done in terms of deviations from the reference values.

The diagram on Fig. 1 shows a general scheme of $\mathcal{TWIST}$ blind analysis. Thin rectangles on the plot represent files on disk, ovals represent different programs run in the course of the analysis.

micheld

Micheld is the program which samples muon decay spectrum and gives GEANT energy and angle of positrons from decays of polarized muons. The values of Michel parameters are thrown at random within given tolerances, encrypted

[1] It was called "base" in [1]. The term "base" had two different meanings in [1]: 1) An opposite of a derivative spectrum, just a "normal" Michel spectrum with any values of Michel parameters. 2) A spectrum which is combined with derivative spectra ($n_i(\varpi)$ in Eq.(22)) to reproduce another spectrum ($n_i(\varpi + \Delta\varpi)$ in Eq.(22)). I'll try to use the term "reference" for the first meaning in this note. However it is still called "base" in `chi2fitter` input file.

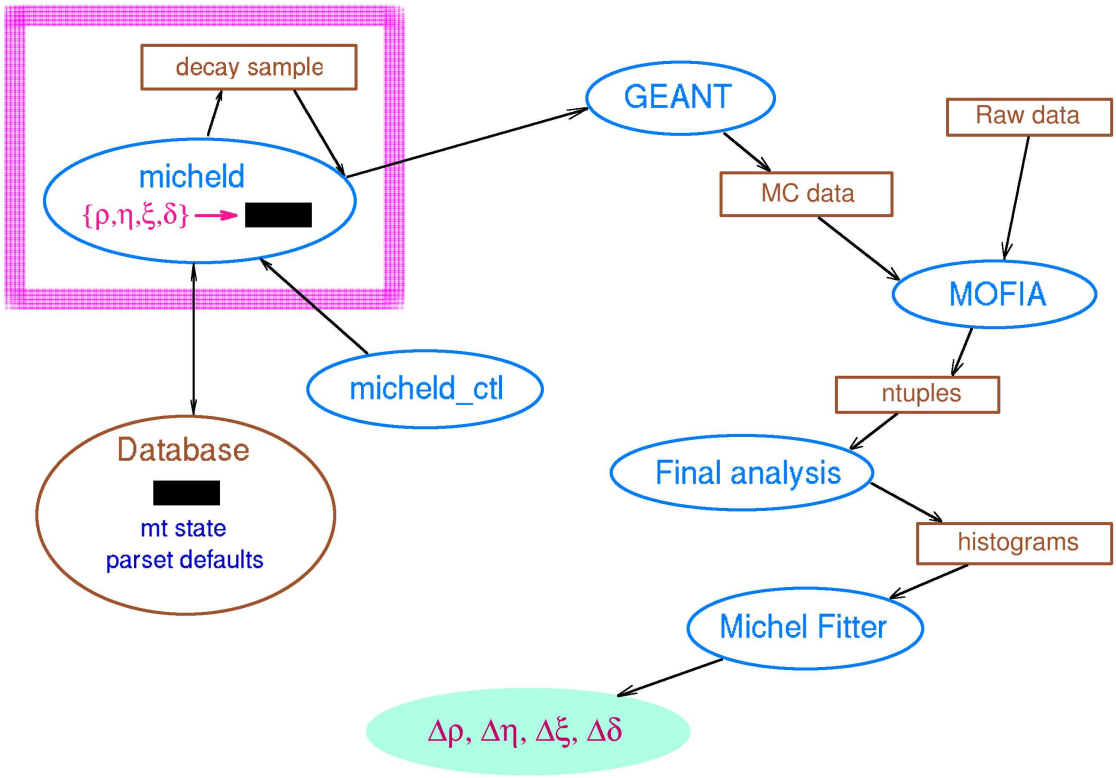

Figure 1: The blind analysis scheme.

using public key cryptography, and stored in a database (the black box on the diagram). Sets of Michel parameters are identified by number ("parset"). Micheld keeps clear text values of parameters in its memory and uses them to generated a requested amount of muon decays. Muon decays are organized into spectra. Spectra may be of different types (base or different derivatives), use different Michel parameters, different radiative corrections, etc. A spectrum consists of a number of samples of fixed size, e.g. 1000 samples of 100,000 decays each. Available spectra are shown on the "Blind analysis database" web page at `http://e614db.triumf.ca/private/michelddb/` (linked from the $\mathcal{TWIST}$ software page). Generated muon decay samples are stored on disk, so that the same spectrum shape can be re-used even after the clear text parameter values are lost due to e.g. machine reboot. Each decay is stored as 3 numbers: energy (double), $\cos(\theta)$ (double), and type and sign information encoded into an integer. Type distinguishes different derivatives and derivatives from base spectra, sign is always positive for base but may be negative for some derivatives. There is also a header for every stored decay sample identifying the sample and containing the number of points thrown in its generation.

The tolerances on Michel parameters and the encryption key are taken from a database. Micheld also checkpoints its state to the database, so that a single thread of random number is used, even between restarts.

Micheld is a multi-threaded TCP/IP server with a compile-time defined limit on the number of simultaneous connections (currently 8).

GEANT

The "standard" $\mathcal{TWIST}$ GEANT has been modified to use muon decays provided by micheld instead of doing its own spectrum sampling. On the first call to a muon decay routine GEANT connects to the server and gets a sample of decays specified in the FFCARD file. Then it uses obtained values of energy and angle to track decay positrons. Due to muon pile-up more than one "black box" decay may be used per GEANT event. When all decays from the sample are used, GEANT run is terminated. So normally you set TRIG to a very large number and let sample size to determine the actual number of events per run[2], thus making sure that the whole sample is used and you get the correct normalization for the fit of Michel parameters.

The sample of muon decays to use is identified by its spectrum number (the second parameter of the SRVU ffcard, Fig. 2) and sample number within the spectrum. The sample number is not given directly; instead it is calculated as RUNG (the GEANT run number) minus IBRUN (the third parameter of the SRVU ffcard). The idea is that to use a range of run numbers starting from N, you set IBRUN to N in you ffcards template, than vary just RUNG starting

[2]It is possible to extend GEANT to use several samples per run instead of just one, if larger files are desirable. The number of samples to use can be a FFCARD parameter.

from N, without making any changes to SRVU. This will make run N to use sample 0, run N+1 sample 1, and so on.

If a connection attempt fails, it is retried as specified in the SRVC ffcard, see Fig. 2.

GEANT writes out decay type and sign for every event, along with MC track number of the muon and decay offset in the sample, into the MCD3 ybos bank. There are also MCB2 and MCE2 banks produced at the beginning and the end of a run. These banks identify the decay sample used and contain the "NTHROWN" number for the sample. The MCE2 bank is produced only *after* the *whole* decay sample has been used. This is a safeguard: if some unused decays are left, NTHROWN is not valid, and use of this run may screw up the values of Michel parameters in the final fit. When MCE2 bank is written out a log file message is also produced:
`NTHROWN =  598090 for request  1041`
so that it is easy to check.

To reproduce an event from the middle of a run for debugging, it is not enough to just set values of the RNDM (or RLX*) ffcard. You should also point GEANT to the right muon decay to use for the event. This is accomplished by setting the IDECAY parameter in the SRVU card. The value of IDECAY is printed by GEANT together with RNDM values. It is also stored in the output file and can be seen in `photo` printout.

MOFIA

Information from MCB2, MCD3, and MCE2 banks is unpacked by MOFIA. When an MCE2 bank is found, the NTHROWN number is printed to the log file:
`unpMCE2(): reqnum =   1041 , nthrown =   598090`
If a log file from a MC analysis does not contain an "unpMCE2" line, something was wrong with either analysis or generation of the run, and it should be ignored in Michel parameter fits.

MCD3 data are passed through MOFIA to the standard ROOT tree. They are also printed out by `photo`.

ntuple analysis

To fill a histogram to be used in the fit, you need 3 numbers per event: reconstructed values of momentum, $\cos(\theta)$, and the "decay sign" giving the weight of the event (± 1);

The sign can be unpacked from the "MichelD_accflag" integer coming from `micheld` and available in the ROOT tree. There are 2 bits per spectrum type,

```
C ========== SRVU: use micheld server  =============
C ISRVUSE,ISPECTRUM,IBRUN,IDECAY
C
C  ISRVUSE>0 to get a sample of Michel decays from a micheld server [0].
C
C  ISPECTRUM  request the given spectrum number [0].
C
C  IBRUN:  Base run number [0]. The sample number used is (RUNG-IBRUN).
C  Samples are numbered from zero within each spectrum.
C
C  IDECAY  is the initial offset in the sample buffer [0].
C  To reproduce an event, set IDECAY as printed out by
C photo in addition to setting random seeds in the RNDM card.
C If the original run had default IDECAY==0, you can also
C calculate IDECAY value to reproduce an event of that run
C as (IEVENT-1), where IEVENT is printed out by GTRIGI.
C
SRVU 1 8 8000 0
C
C ========== SRVC: micheld connection parameters  =============
C ip1 ip2 ip3 ip4 port ntries delay_min delay_max
C
C Server address is:  ip1.ip2.ip3.ip4:port
C The default is 142.90.100.68:3469, that is, lin00.triumf.ca:3469
C
C Up to ntries [1] connection attempts are made. The time interval
C between two connection attempts starts at (randomized) delay_min [5s]
C and doubles (with randomization) till it gets to delay_max [1000s],
C then it stays around delay_max.
C
C nunatak1
SRVC 192 168 0 11   3469   8 30 1000
C lin00:
C SRVC 142 90 100 68   3469   9 30 1000
C
```

Figure 2: FFCARD parameters

with the '01' pattern representing the '+' sign, and '11' representing '-'. The '10' pattern is not used and should never happen, and '00' means the given spectrum type was not accepted for the decay[3]. So, the sign for a "sptype" spectrum can be obtained as following

```
sign = 2 - (0x3 & (accflag>>(2*sptype))),
where sptype is one of
    0    base
    1    rho
    2    eta
    3    xi
    4    delta
    5    xidelta
    6    xixidelta  (xi in the linear parametrization)
```

the value of the expression may be ± 1, or 2. In the latter case the decay does not belong to the "sptype" spectrum. Since combined spectra with several derivatives are not used in production, instead of using the expression above you can just enumerate the possibilities:

```
switch(accflag) {
    case 0xc: case 0x30: case 0xc0:
    case 0x300: case 0xc00: case 0x3000:
      weight = -1;
      break;

    case 0x1: case 0x4: case 0x10: case 0x40:
    case 0x100: case 0x400: case 0x1000:
      weight = +1;
      break;

    case 0: // ignore event-decay cache exhausted
      weight = 0;
      break;

    default:
      ERROR, stop here.
}
```

Then you fill the histogram using *ptot*, *costh*, and *weight*:
`CALL HFILL(hid, ptot, costh, weight)`, (HBOOK) or
`hid->Fill(ptot, costh, weight)` (ROOT).
It is important to make sure errors are calculated correctly at the fill time.

[3]It is possible for micheld to generate a combined spectrum with several derivatives at the same time, this is why the "accflag" definition is a bit more complicated than a plain ± 1.

This in not the default!!! Also note that you can not use the HF2() call since it does not do error calculation! To enable error calculation in HBOOK, call `HIDOPT(hid, 'STAT')` once on the spectrum histogram after booking but before filling. In ROOT, call `hid->Sumw2()` once, also before filling.

Muon pileup

Supposedly only events with a single reconstructed muon decay are accepted for the final spectrum. But it is possible that a "good" reconstructed event in fact contain more than one muon decay in its MC banks. Which of the several `accflags` should we use to calculate the sign of the event then?

Usually it is possible to correlate the reconstructed decay with just one of decays in the MC banks. (In most of the accepted double muon decay events the first muon hits dead material upstream and is lost at large negative z, and the other one has its decay vertex close to the target, so the choice is obvious.) Then we can use `accflags` for the "right" decay. In my tests I used z and vertex time to do the correlation, however the cuts were not tuned and probably were not optimal, so this question requires more investigation.

Note that for a base spectrum the weight should be always positive, so the ambiguity does not matter here.

chi2fitter

There is a Web page explaining how to compile the fitter
`http://e614db.triumf.ca/private/offline/mcfitter/mcfitter.html`
linked from the Software page. By default the fitter support reading of ROOT files only. You can easily enable reading of HBOOK files as well by editing the Makefile, see the web page.

The fitter takes one command line argument which is a name of a "fit specification file". There is an example file in the fitter source directory, its contents is shown on Fig. 3.

CUTS are the fiducial volume cuts,

$$[xmin, xmax] \times [acmin \leq |\cos(\theta)| \leq acmax].$$

DATA is the spectrum being fit, BASE is the reference spectrum. A spectrum is specified by its file name and histo ID in the file (an integer if you use HBOOK, the example is for ROOT files).

EFF is the spectrum used for normalization. The first number is NTHROWN for the spectrum, that there is a histo specification like for DATA and BASE.

There is one DERIV spectrum for each fit parameter. The first entry is the name of the parameter, than there is NTHROWN and histo specification.

```
# Comments are allowed,

# as well as emty lines. The order of lines is not important, but
# CUTS, DATA, BASE, EFF, and at least one DERIV should be present.

CUTS xmin=15.85 xmax=51.5 acmin=0.5 acmax=0.95

DATA   data.root base2  # an endline comment

BASE   base.root base2

# The number is Nthrown
EFF        596293709   eff.root  base2

# Name each derivative and give Nthrown before the file name and histo id.
DERIV rho  105636497   rho.root  rho2
DERIV eta  3076006601   eta.root  eta2 # another endline comment
DERIV xi   318545968   xi.root   xi2
DERIV delta 215619755   delta.root delta2

# comment out the OUTFILE record to avoid writing the file
OUTFILE     fittest.root

#EOF
```

Figure 3: Fit specification file

Note that input spectra can be in different files, or all or some of them may be in the same file.

When the fitter is run, the usual ROOT MINUIT output is produced, then the final result is printed, like

```
Fit result converted to the xi,delta parametrization:
chi2=2270.2, ndf=2314, confLevel=0.738489
  rho = -0.014709 +-  0.002047
  eta = -0.105225 +-  0.114123
   xi =  0.059175 +-  0.002028
delta = -0.011836 +-  0.001053

Correlation coefficients:
             rho        eta         xi      delta
  rho   1.000000   0.949402   0.739808   0.129513
  eta   0.949402   1.000000   0.741554   0.102415
   xi   0.739808   0.741554   1.000000  -0.345164
delta   0.129513   0.102415  -0.345164   1.000000
```

If there are fit parameters called "xixidelta" and "xidelta" they are converted to "xi" and "delta" before the printout. The conversion is discussed in the Appendix.

If an OUTFILE is defined in the fit specification file, fit residuals and some other histograms are written out.

Here is a way to calculate NTHROWN. First of all, check that the runs are OK. (Assume log files of your analyzed runs are finished/*/*.txt)

```
bash
for i in finished/*/*.txt; do grep -q 'nthrown' $i || echo $i; done
```

Problem runs are echoed by the command, no output means everything is OK. Now do the calculation:

```
awk '/nthrown/{s+=$NF; print s"\t"$0}' finished/*/*.txt
```

gives

```
595939   unpMCE2(): reqnum =   41 , nthrown =   595939
1191579  unpMCE2(): reqnum =   42 , nthrown =   595640
1785830  unpMCE2(): reqnum =   43 , nthrown =   594251
....
```

The first number on the last line is the answer.

Systematics

To study systematic effects while keeping analysis blind to the values of Michel parameters, the following method can be used: a Michel spectrum obtained for some "standard" conditions is compared to another spectrum, which is produced with a controlled change in MC, or reconstruction, or data taking parameters.

To fit Michel parameters, one needs a "data" spectrum (the one to be fit), a "reference" ("BASE") spectrum, a set of "derivative" spectra, and a "normalization" ("EFF") spectrum to get the right relative weight for the reference spectrum. "Data" and "reference" can be either real data, or MC. "Derivative" and "normalization" spectra ought to be made from Monte-Carlo. When the reference spectrum is a Monte-Carlo one, it can also be used for normalization. It is crucial that reference, normalization, and derivative spectra are all generated and analyzed in exactly the same way.

Below are suggested steps to get the required histograms for several different classes of systematic studies. The standard ntuples and/or GEANT runs [2] should be used as much as possible to avoid the waste of CPU time. The "reconstruction effects" are understood broadly here; e.g. all the calibrations are included in this category (since it is the reconstruction software that uses calibrations).

1. **Reconstruction systematics, using MC**

   One can use a "standard" set of ntuples for the reference and all derivative spectra. See [2] for information about current standard samples.

   Retrieve from CVS the version of MOFIA documented in [2], also get the kcm file used for the "standard" analysis. Make the change you want to study to the code and/or the kcm file. (E.g., use alignments different from what the MC was generated with, or change a windowing parameter, or ...)

   Run the modified analysis on the second base set of standard MC runs, use it to prepare your "data" histogram for the fit. The normalization spectrum (the "EFF" in the fit specification file) can be the same as the base.

2. **Monte-Carlo systematics**

   Similar to case (1), but here you'll need to produce your own MC samples as well.

   Get the CVS version of GEANT and ffcard file documented in [2]. Introduce the change you want to look at. Generate your high-statistics data set using the same black box spectrum number as the second base in standard MC. Get the version of MOFIA and KCM file from [2], use it to analyze the runs you have produced. Make your "data" spectrum. Use

standard ntuples for reference, derivative, and normalization (the same as the reference) spectra.

3. **Data to data, different running conditions**

   Here we are looking at a difference in the spectrum shaped cause by different running conditions (high voltage, atmospheric pressure, beam intensity, ...). One of the two data sets is used to produce the reference spectrum, the second one is being fit.

   Note: To evaluate an effect of the change on Michel parameters, Monte-Carlo derivatives and normalization still have to be used. The Monte-Carlo settings should correspond to the data taking conditions of the reference data set. There is an assumption that GEANT has been validated and does reproduce the data, otherwise the fit may not be meaningful.

   Retrieve from CVS the version of MOFIA documented in [2], also get the kcm file used for the "standard" analysis. Make no changes, analyze both data sets in the same way. Prepare your "data" and "reference" histograms.

   Derivative and normalization spectra can be made from the standard ntuples.

4. **Data to data, reconstruction effects**

   The systematic effects of the reconstruction can be studied by analyzing uniform experimental data in two ways and looking at the difference.

   Retrieve from CVS the version of MOFIA documented in [2], also get the kcm file used for the "standard" analysis. Use this setup to analyze a number of runs. Prepare your "reference"spectrum.

   Make the change you want to study to the code and/or the kcm file. (E.g., use a different cross-talk removal algorithm.) Analyze some *other* data runs, taken under the *same conditions*. Prepare the "data" spectrum.

   Derivative and normalization spectra can be made from the standard ntuples. Monte-Carlo should be validated and correspond to the data used, see a note in the "Data to data" case.

Appendix

Here are the formulas for conversion between the ξ, δ, covariance matrix V, and $z = \xi|_{\xi\delta}, w = \xi\delta$, covariance matrix U, parametrizations.

1) $\xi, \delta \longrightarrow \xi\delta$

$$\begin{aligned} \Delta z &= \Delta\xi \\ \Delta w &= (\xi_0 + \Delta\xi)(\delta_0 + \Delta\delta) - \xi_0\delta_0 \\ U_{ij} &= V_{ij}, \qquad i, j \neq w \end{aligned}$$

$$
\begin{aligned}
U_{iw} &= (\delta_0 + \Delta\delta)\, V_{i\xi} + (\xi_0 + \Delta\xi)\, V_{i\delta}, \qquad i \neq w \\
U_{ww} &= (\delta_0 + \Delta\delta)^2\, V_{\xi\xi} + 2(\xi_0 + \Delta\xi)(\delta_0 + \Delta\delta)\, V_{\xi\delta} + (\xi_0 + \Delta\xi)^2\, V_{\delta\delta}
\end{aligned}
$$

2) $\xi\delta \longrightarrow \xi, \delta$

$$
\begin{aligned}
\Delta\xi &= \Delta z \\
\Delta\delta &= (\Delta w - \delta_0 \Delta z)/(\xi_0 + \Delta z) \\
V_{ij} &= U_{ij}, \qquad i,j \neq \delta \\
V_{i\delta} &= -\frac{\xi_0\delta_0 + \Delta w}{(\xi_0 + \Delta z)^2}\, U_{iz} + \frac{1}{\xi_0 + \Delta z}\, U_{iw}, \qquad i \neq \delta \\
V_{\delta\delta} &= \frac{(\xi_0\delta_0 + \Delta w)^2}{(\xi_0 + \Delta z)^4}\, U_{zz} - 2\,\frac{\xi_0\delta_0 + \Delta w}{(\xi_0 + \Delta z)^3}\, U_{zw} + \frac{1}{(\xi_0 + \Delta z)^2}\, U_{ww}
\end{aligned}
$$

The covariance matrix conversion is approximate, see e.g. [3] for details.

Note that the value of $\Delta\delta$ depends on the reference values ξ_0, δ_0 used for the fit. In the blind analysis technique those values are not known exactly; however we know that $\xi_0 = 1 \pm 0.03$, $\delta_0 = 0.75 \pm 0.02$, that gives about 5% precision on the *deviation* $\Delta\delta$. This is sufficient for systematic studies: it gives a 5% uncertainty on sensitivities to different systematics and a 5% uncertainty on the estimated systematic error on δ. Other parameters, including ξ, are not affected by this uncertainty. In the fitter, conversion function is called with arguments $\xi_0 = 1$, $\delta_0 = 0.75$, after the "black box" is open these numbers can be adjusted to find the true value of $\Delta\delta$.

References

[1] A.Gaponenko, *"Monte-Carlo fitting: the spectrum expansion"*, TWIST Technical Note #80, April 2003.

[2] TWIST Official Monte-Carlo samples, `http://e614db.triumf.ca/private/mcsamples/`

[3] W.T.Eadie, D.Drijard, F.E.James, M.Roos, B.Sadoulet, *"Statistical methods in experimental physics"*, North-Holland, 1971.