

Table of Contents

Fit Page	1
B to hh Page.....	1
Documentation.....	1
Fit Strategy.....	1
Fit Code.....	1
How to get the code.....	1
Informations on the recommended tags.....	1
How to compile/run the code.....	2
Current Fit options.....	3
Fit configuration.....	3
Parameters configuration.....	4
Fit Studies.....	4

Fit Page

These pages contain the information about the B2hh CP Fit. Detailed information about the technical issues related to the fit implementation are discussed.

B to hh Page

Documentation

An useful overview of technicalities related to Likelihood fits can be found in this very nice page [maintained](#) by G.Raven. This page contains as well code examples for various fits. Detailed documentation on B2hh Fit code is contained in this web page.

Fit Strategy

The B to hh events selected as outlined in here [are](#) used in a combined mass,time fit to extract the direct and mixing CP asymmetries. All the 8 B to hh data samples are fitted simultaneously: the PID information can be either used to select different samples or directly into the fit (see below in the fit documentation how to perform simultaneous or separate fits). The resulting CP informations are combined using U-spin symmetry in order to measure the CKM angle gamma.

Fit Code

The fit code has been written using the RooFit [package](#) provided in ROOT [package](#). The unbinned extended maximum likelihood fit is performed on the combined PDF of mass and time for signal and background events.

How to get the code.

The Fit code is kept under a cvs project. The project can be browsed by looking in this [page](#). To get the code you should follow the instructions below.

- First of all you should set the CVS variables. Example below is given for the tcsh shell.
 - ◆ Outside cern/lxplus
 - ◇ setenv CVS_RSH ssh
 - ◇ setenv CVSROOT :ext:isscvcs.cern.ch:/local/reps/B2hhFit
 - ◆ On lxplus
 - ◇ setenv CVSROOT :kserver:isscvcs.cern.ch:/local/reps/B2hhFit
- Then you can get the tag you want.
 - ◆ cvs co -r <tagname> B2hhFit [page](#)

Informations on the recommended tags

Detailed information on available tags is given below:

- v6r0 : **Latest tag. Recommended.** . Contains the code of the Sim Fit validated.
- v5r3 : Contains: Fixed approach for tagging categories. Different per event proper time error for signal and background. Priors on CP parameters have been removed. We did left ONLY a prior on the mistag.
- v5r2 : First tag with working sim fit. Requires patched code.
- v5r0 : This tag can be used to perform separate Bs and Bd fits. Does not require patched code.

Perform variable plots and likelihood scans. Sim fit still does not work.

- v3r4 : First fit version that implements per event Proper time error on separate fits. Sim fit does not work.

How to compile/run the code.

Instructions on how to compile the code are given below:

- On 'your' laptop/desktop
 - ◆ First of all you should check for libRooFit.so, libMathCore.so and libMathMore.so in your root library distribution. If you do not have them you should recompile ROOT having enabled those libraries using the `./configure --enable=roofit --enable=mathcore --enable=mathmore` syntax.
 - ◆ Then you properly set the ROOT variables:
 - ◇ ROOTSYS : should point to your local ROOT installation
 - ◇ LD_LIBRARY_PATH should point to \$ROOTSYS/lib
 - ◇ PATH should contain \$ROOTSYS/bin
 - ◇ Then you compile ROOT by: `./configure --enable=roofit; make; make install`
 - ◆ Then you are ready to compile the code issuing the command 'gmake FitCode' in your B2hhFit directory
- On lxplus
 - ◆ First of all you should set directly the ROOT environment variables:
 - ◇ `setenv ROOTSYS /afs/cern.ch/sw/lcg/external/root/5.18.00/slc4_amd64_gcc34/root`
 - ◇ `setenv LD_LIBRARY_PATH`
`./$ROOTSYS/lib:/afs/cern.ch/sw/lcg/external/GSL/1.8/slc3_ia32_gcc323/lib` . **You need to add also the path to your working directory!**
 - ◇ `set path=($ROOTSYS/bin $path)`
 - ◆ Then you're ready to compile the code using 'gmake FitCode'
- On CNAF site
 - ◆ You just need to set properly the ROOT variables and then you're ready to compile.
 - ◇ `setenv ROOTSYS /opt/exp_software/lhcb_user/software/root/v5.18.00/root`
 - ◇ `setenv LD_LIBRARY_PATH`
`/${ROOTSYS}/lib:/afs/cern.ch/sw/lcg/external/GSL/1.8/slc4_ia32_gcc34/lib` . **You should add also the path to your working directory**
 - ◇ `set path=($ROOTSYS/bin $path)`
 - ◇ Then you're ready to compile the code using 'gmake FitCode'

Instructions on how to run the code are given below. The B2hhFit package contains the directory 'tools' : you can find there some useful perl scripts that can be used to submit jobs for different use cases. The package contains also a 'ganga' directory that contains templates and a macro for job submission on the grid via ganga and Dirac. The details of the three typical user cases are listed below, as well as a quick description of the two perl script that can be used to run fits or systematics studies.

- Before starting.
 - ◆ The job submission on local or batch machines is handled by a perl script named 'SubFit.pl'. This script needs to be properly customized to your code configuration in order to work properly. The customization can be done by editing the script and settings the following variables to your specific needs. **Note that all paths need to be absolute path in order to be properly understood when running batch jobs!**
 - ◇ \$scratch = should point to a directory where you want to dump the output
 - ◇ \$softdir = should point to the directory where the B2hhFit fit code is kept
 - ◇ \$optdir = should point to the directory where you want to dump the option files used to run a given fit
- Running jobs on local machine (interactively)

- ◆ To run a single job you can try the following syntax: tools/SubFit.pl -int -dl -run BD -flag myTestDb
- Submitting jobs to LSF
 - ◆ To run a single job under a batch system (lxplus or cnaf) you can try the following syntax: tools/SubFit.pl -q <queue> -run BD - flag myQueueBd
- Submitting jobs to LCG (using ganga, Dirac)
 - ◆ First of all you need to customize the ganga/MyRooFit.txt template file to your code setup.
 - ◆ After that you log into ganga, then you 'load' the template MyRooFit.txt and you submit the job.

Current Fit options

The fit configuration is kept in two different files. The first one: config/myConfs.txt contains the instructions on which type of fit you're going to run. The second one config/config.txt contains the default values for the parameters that are going to be used in the fit.

Fit configuration

- Running configurations. Main configuration (input directories, output directories, output flagging, signal + background approach, time and mass approach, analytical integration) can be performed in the config/myConfs.txt file:
 - ◆ Fit strategy (perform a simple mass fit or a combined mass*time fit) is controlled by the flag : doTimeAndMass [0 = only mass fit, 1 mass*time fit (default)]
 - ◆ Fit strategy (perform a simple signal events fit or adding also bkg events) is controlled by the flag : doSigPIBack [0 = only signal evts, 1 sig + bkg (default)]
 - ◆ Analytical integration is controlled by the flag: doAnaPdf [0 = use RooFormulaVar and RooGenericPdf, 1 use compiled class (default)]
 - ◆ The generation of control plots (variables and likelihood scans) is controlled by the variable: doPlots [0 = no plots are created, 1 plots are made (default)]
 - ◆ The execution of the fit is controlled by the flag : doFit [0 = no fit performed. Only data is generated (if specified), 1 fit is performed (default)]
 - ◆ The printout level is controlled by the variable: debugLevel [0 = lowest printout level, 1 debugging information is added (default)]
 - ◆ Output files can be flagged with the string provided set in **#outFlag** <string> variable
 - ◆ Input files have path relative to the one specified in the %RED **#inDir** <path> variable.
 - ◆ Input files have path relative to the one specified in the %RED **#outDir** <path> variable.
- Different Fit approaches are implemented [to configure this option using the perl script use the -run option]:
 - ◆ Running a Bd fit: in this case only the PDFs for Bd decays are going to be used in the fit. Three different data set are generated/read and used to build three likelihood variables that are simultaneously minimized. To select this option you should set the #runMode variable to **BD** in the file myConfs.txt
 - ◆ Running a Bs fit: in this case only the PDFs for Bs decays are going to be used in the fit. Three different data set are generated/read and used to build three likelihood variables that are simultaneously minimized. To select this option you should set the #runMode variable to **BS** in the file myConfs.txt
 - ◆ Running a simultaneous fit: in this case all the PDFs for Bd and Bs decays are going to be used in the fit. Only one data set is generated/read and used to build the likelihood variable that is minimized. To select this option you should set the #runMode variable to **Sim** in the file myConfs.txt
- Different Proper time resolution models
 - ◆ Gaussian resolution: to use this approach you should set **doTimeResolution** variable to **0** in config/myConfs.txt file. [default option]

- ◆ Gaussian resolution with per event error: to use this approach you should set **doTimErPerEv** variable to **1** in config/myConfs.txt file. The information on the per event error is used to generate a RooHistPdf and can be taken from:
 - ◇ a generated Gaussian distribution: to use this approach you should set **doProvidePEe** variable to **1** in config/myConfs.txt file. Mean and sigma can be set in the config/config.txt file.
 - ◇ a generated Landau distribution: to use this approach you should set **doProvidePEe** variable to **2** in config/myConfs.txt file. MPV and 'sigma' can be set in config/config.txt file.
 - ◇ a provided file: to use this approach you should set **doProvidePEe** variable to **0** in config/myConfs.txt file. [default option] File name should be "data/B2pipiPTerr.root".
- Different data output
 - ◆ Data set can be 'persisted' into root files or ascii files
- Different mistag approaches:
 - ◆ Fitting a mistag parameter inside the RooBDecay model [default option]. The mistag parameter can be set in the config/config.txt file.
 - ◆ Providing a mistag information: to use this approach you should set **doOmegaPerEv** variable to **1** in config/myConfs.txt file. The file that contains the omega information can be specified by setting the variable #omeFile to **data/omega.txt** in the file config/myConfs.txt .
 - ◆ Performing a fit in tagging categories.

Parameters configuration

Fit Studies

- Tagging studies
 - ◆ M. Calvi talk [↗](#) at the Flavour Tagging Meeting 06/09/06
- Proper time studies

-- Main.asarti - 06 Jul 2007

This topic: LHCb > FitPage

Topic revision: r9 - 2008-05-13 - AlessioSarti



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback